

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 12-09-2012		2. REPORT TYPE Final		3. DATES COVERED (From - To) 01/03/2009 - 30/09/2011	
4. TITLE AND SUBTITLE CHARACTERIZATION AND PLANNING FOR COMPUTER NETWORK OPERATIONS				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER FA9550-09-1-0101	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) George Cybenko				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Dartmouth College Hanover NH				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research Suite 325, Room 3112 875 Randolph Street Arlington, VA 22203-1768				10. SPONSOR/MONITOR'S ACRONYM(S) AFOSR	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-OSR-VA-TR-2012-1091	
12. DISTRIBUTION / AVAILABILITY STATEMENT Distribution A: Approved for Public Release					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Whom we e-mail, where we browse, what we purchase, and the things we search for on the World Wide Web all leave identifiable traces of who we are as individuals. In today's technology focused landscape, cyberspace represents the new environment in which we communicate, work, shop, and play, and the cyber behaviors we exhibit there give a great deal of insight into our individual identities. This dissertation proposes a novel approach to the modeling and analysis of behaviors based on a user's cyber activities. We present a methodology to identify, extract, and analyze cyber behaviors providing the foundation for cyber-based behavioral modeling. In addition, we define the underpinnings necessary to support this approach through our behavioral extraction, Bayesian sample size estimation, and behavioral state-based techniques, then empirically evaluate their use. Methods are implemented to characterize, predict, and detect change in individual and group behaviors and we demonstrate their effectiveness using real world data.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 232	19a. NAME OF RESPONSIBLE PERSON George Cybenko
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (include area code) 603-369-1133

Cyber-Based Behavioral Modeling

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of

Doctor of Philosophy

by

David John Robinson

Thayer School of Engineering
Dartmouth College
Hanover, New Hampshire

July 2010

Examining Committee:

Chairman_____

Dr George Cybenko

Member_____

Dr Stephen Taylor

Member_____

Dr Vincent Berk

Member_____

Dr Kamal Jabbour

Brian W. Pogue

Dean of Graduate Studies

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

Abstract

Whom we e-mail, where we browse, what we purchase, and the things we search for on the World Wide Web all leave identifiable traces of who we are as individuals. In today's technology focused landscape, cyberspace represents the new environment in which we communicate, work, shop, and play, and the cyber behaviors we exhibit there give a great deal of insight into our individual identities.

This dissertation proposes a novel approach to the modeling and analysis of behaviors based on a user's cyber activities. We present a methodology to identify, extract, and analyze cyber behaviors providing the foundation for cyber-based behavioral modeling. In addition, we define the underpinnings necessary to support this approach through our behavioral extraction, Bayesian sample size estimation, and behavioral state-based techniques, then empirically evaluate their use. Methods are implemented to characterize, predict, and detect change in individual and group behaviors and we demonstrate their effectiveness using real world data.

This research offers valuable contributions to a number of areas. Within the Department of Defense, this work provides the foundation to accurately and effectively represent and analyze the behavioral layers of the cyber situational awareness environment, which in turn will provide planners and decision makers critical information to achieve their mission. In addition, the financial sector may leverage behavioral models for fraud prevention and credit scoring while human resources can track employees by monitoring their behavior for malicious activity (insider threat) and for classifying and ranking user's skills. E-commerce can use these techniques to further expand the scope of its profiling techniques to best characterize and predict purchase patterns for individuals and groups of online shoppers as well.

Dedication

I dedicate this Doctoral dissertation to my late mother Jonna Robinson and to my father William Robinson. None of this would be possible without the work ethic you instilled and the support you provided over the years.

Acknowledgements

I would like to thank Professor George Cybenko for everything. For getting me to Dartmouth, providing mentorship, guidance, and support, and for putting up with a forty year old student who has been out of academia for one or two years. You are a tremendous individual and I will miss working for you. What else can be said. You are a “rock star”. I would also like to thank the rest of my committee, Professor Stephen Taylor, Dr Vincent Berk, and Dr Kamal Jabbour for their time, comments, advise, and occasional “beer lunch” professional development (that would be with Dr Berk for those unsure).

I would like to thank all members of the research group for their time, patience, and willingness to teach an old dog new tricks. I know you will be hard pressed to keep the conversations at the same intellectual level in my absence. You guys are great and I wish you all the best.

Most importantly, I would like to thank my wife Andrea and children Ellie and Aiden. To Ellie and Aiden, I thank you for knowing when to give hugs, knowing when to run away, and knowing there is no “I” in team, but there are two of them in Wii. To Andrea, the second foremost expert on cyber-based behavioral modeling, thank you for yet again putting up with, supporting, and proof reading like no one else could. You are amazing.

Contents

Abstract	iii
Dedication	iv
Acknowledgements	v
List of Figures	x
1 Introduction	1
1.1 Cyber Situational Awareness	4
1.2 Behavioral Modeling Framework	8
1.3 Contribution of Work	14
1.4 Structure of Thesis	16
2 Previous Research on Cyber-Behavior Analysis	18
2.1 Web Usage Mining	18
2.2 Engineering	27
2.3 Government/Industry	30
2.4 Psychology	36

3	Cyber Behavior Observables	39
3.1	Online Data	40
3.2	Offline Data	43
3.3	Pre-Processing	45
3.4	Summary	52
4	Behavioral Activities	54
4.1	Activity Ontology	55
4.2	Activity Assignment	64
4.3	Activity Inputs	76
4.4	Activity Assignment Accuracy	79
4.5	Summary	82
5	Behavioral Analysis	83
5.1	Behavioral Model	83
5.2	Behavioral Traits	91
5.3	Behavioral Sample Size Estimation	93
5.4	Behavior-Based Characterization	99
5.5	Prediction	117
5.6	Anomaly Detection	126
5.7	Summary	130
6	Empirical Evaluation of Cyber-Based Behavioral Models	131
6.1	Datasets	131
6.2	Use Cases	140

6.3	Summary	180
7	Future work and Conclusions	181
7.1	Future Work	181
7.2	Conclusions	184
A	Tools and Code Resources	185
A.1	Apache Commons Math	185
A.2	CLUTO	185
A.3	Java	186
A.4	Java Universal Network Graph (JUNG)	186
A.5	JFreeChart	186
A.6	Lucene	186
A.7	Mallet	187
A.8	MySQL	187
A.9	R	187
A.10	Rserve	188
A.11	Weka	189
B	DMOZ Top Level Category Descriptions	190
B.1	Adult	190
B.2	Arts	190
B.3	Business	191
B.4	Computers	193
B.5	Games	194

B.6 Health	195
B.7 Home	195
B.8 Kids and Teens	195
B.9 News	195
B.10 Recreation	196
B.11 Reference	196
B.12 Regional	196
B.13 Science	196
B.14 Shopping	197
B.15 Society	197
B.16 Sports	198
B.17 World	198

List of Figures

1.1	Table depicting kinetic and non-kinetic characteristics, highlighting the relative immaturity of cyber operations compared to their kinetic counterparts. .	2
1.2	Quad chart representing the core areas needed for effective cyber operations.	3
1.3	Depiction of the geographic aspects of the cyber situational awareness model's <i>physical</i> layer.	5
1.4	Depiction of the communications and infrastructure aspects of the cyber situational awareness model's <i>physical</i> Layer	5
1.5	Updated view of the cyber situational awareness model with the <i>informational</i> layer added.	5
1.6	Graphical depiction of the mediums used to interact with the cyber environment.	8
1.7	Graphical depiction of the interactions and groupings which occur in the cyber environment.	8
1.8	Hierarchical activity tree for <i>Shopping</i> . Sibling nodes in the diagram are not in and of themselves independent activities, but rather a more detailed description of the <i>root</i> activity (i.e. <i>Automobiles</i> is not an activity, but <i>Shopping/Vehicles/Automobiles</i> is).	9

1.9	Independent activity systems for a user. Each large colored oval represents the system (<i>Shopping, Hobbies, Work</i> , etc.) associated with an activity, while the overlaid line diagram denotes the temporal aspects of each lower level node (i.e. how often the user shops for cars, clothes, etc.).	11
1.10	System of Systems representation showing all possible interactions among activity systems of a user. The large blue oval represents the behavioral environment.	12
1.11	User’s “car buying” behavior extracted from the SoS diagram of Figure 1.10.	13
1.12	Graphical representation of our behavioral modeling methodology.	14
2.1	Graphical summary of the impediments to using server-side data.	23
2.2	Social network diagram demonstrating how media reports can be used to create a detailed network of the terrorist organization associated with the September 11 attacks against the World Trade Center.	32
3.1	Overview of the online and offline observables, pre-processing steps performed on each, and their use in this dissertation.	39
3.2	Graphical representation of the input phase of our behavioral modeling methodology.	40
4.1	Knowledge discovery phase of our behavioral modeling methodology.	55
4.2	Generic (left hand side) and ontology instantiated (right hand side) representation of the Lucene index structure.	68
4.3	Graphical depiction of our index instantiation using DMOZ and blacklist data.	69

4.4	k -NN algorithm for k equal to six (solid circle) and k equal to eight (dashed circle).	71
4.5	Selection of the most relevant category from our <i>Dartmouth College</i> example with a confidence interval of 80% and a $K_threshold$ value set to 0.2 (we want to be 80% confident we are within 20% of the true mean for each category). .	75
4.6	Modified k -NN algorithm based on Bayesian inferencing	76
4.7	Data flow diagram depicting how URLs are given activity labels.	77
4.8	Accuracy of using manually generated keywords (DMOZ) versus human generated Delicious tags.	81
5.1	Graphical representation of the analysis phase of our behavioral modeling methodology.	84
5.2	Mapping of the topics model concept of documents being a mixture of topics to our behavioral model of behaviors being a mixture of sessions.	84
5.3	Graphical depiction of the generative process (left) and that of statistical inference (right) as it relates to the generation and extraction of cyber behaviors. .	85
5.4	Session/activity matrix representation of an individual's cyber activities. . .	86
5.5	The matrix factorization of a session/activity matrix into a behaviors/activity matrix and a sessions/behavior matrix	87
5.6	Plate notation depicting our cyber-based behavioral model	89
5.7	Depiction of two 1 st order models showing total activity counts (left) and conditional frequencies of activities (right)	92
5.8	Depiction of a 2 nd order Markov model showing the probabilities of performing an activity given an activity was just performed.	93

5.9	Data flow diagram of adaptive control mechanism used to dynamically grow (left branch of decision node) or shrink (right branch of decision node) the sliding window size.	98
5.10	User profile making use of temporal and contextual attributes to identify those browsing <i>News</i> in the AM and PM hours	104
5.11	Number of dimensions required to represent 1,092 users with URLs and various levels of activity labels.	107
5.12	Determination of naïve sample size by iteratively removing 10% of data and evaluating classifier accuracy using 10 fold cross validation. Using 100% of the data yields the highest (87%) accuracy.	108
5.13	Evaluation of fingerprinting technique using a 90/10 train/test split on click-link URLs and various levels of activity descriptions.	109
5.14	Evaluation of fingerprinting technique using empirical samples size estimates on click-link URLs and various levels of activity descriptions.	109
5.15	Evaluation of fingerprinting technique using empirical samples size estimates and multinomial naïve Bayes classifier on top three activity labels.	110
5.16	Demographic information returned from Quantcast for www.dartmouth.edu	116
5.17	Plot of total counts per session (left) and proportion of sessions per session	119
5.18	Inter-state prediction (solid blue line) thirty five sessions into the future with solid purple line depicting actual activity.	122
5.19	Plot of the predictability of the behavioral states of Figure 5.18. As the number of sessions in the state increase, so does the predictability of future behaviors occurring in the same state.	123

5.20	Markov model depicting user with a clearly persistent behavior (left) and a fairly unstable behavior (right) over time.	123
5.21	Initial intra-state transition matrix prior to any observations. Transitions are based on the assumption that a persistent state will stay in a persistent state and a transient state will transition to any other transient state with equal probability.	124
5.22	Simple two state Markov model used to demonstrate steady state behavioral calculations.	125
5.23	CUSUM chart showing an “in-control” process up to time 20. The remaining 10 time units show the impact of a small increase ($\hat{\sigma}/2$) on the upper CUSUM.	128
6.1	1 st order model depicting the top level categories visited by 12 users.	134
6.2	Listing of the top ten full categories and associated counts for the “browser history” users.	134
6.3	Comparison of national average browsing characteristics to our 12 “browser history” users.	135
6.4	1 st order model of the top level categories for 4,188 selected AOL users.	138
6.5	Top 20 activities performed by 4,188 AOL users.	139
6.6	Sessionization histogram for 4,188 AOL users. Histogram emphasizes 82% of the population had 125-215 query sessions.	140
6.7	The goal of this scenario is to affect the environment in some way and then determine if/how the user’s behavior changes.	142
6.8	Department of Defense Joint Targeting Cycle	143

6.9	Graphical depiction of the key components of an Integrated Air Defense System (IADS). The command, control, and communications component (highlighted in red) represents the target of interest for this scenario.	144
6.10	Phase five of the Joint Targeting Cycle; Find, Fix, Track, Target, Engage, and Assess.	145
6.11	Plot of user 1 profile characteristics over stable sessions with a superimposed least squares regression plot (dashed red line).	147
6.12	1 st order model of all top level activities for user 1. Size of activity nodes is based on the percentage of the root node represented (i.e. larger the node, larger the percentage)	148
6.13	User 1 1 st order model for the <i>Computer</i> activity.	149
6.14	User 1 1 st order model for of <i>Software</i> and <i>Programming</i> activities.	150
6.15	Five behaviors associated with user 1 extracted using LDA. Behaviors shaded green represent work related behaviors while those colored red are non-work or leisure related behaviors.	151
6.16	Plot of the proportion of time user work behaviors are exhibited over sessions for User 1 (red line) with a superimposed least squares fit (dashed black line). Upper and lower confidence bounds depicted as dashed blue and green lines respectively.	153
6.17	1 st order histogram of the proportion of user 1's queries made over a twenty four hour period.	154
6.18	Plot of cumulative differences from baseline fingerprint proportions for each category for User 1. Relative lack of deviation indicates the individuals behavior did not change significantly during this time period.	155

6.19	The goal of cyber stress monitoring is to detect and identify stimulation to the environment from behavioral changes to individuals and groups.	156
6.20	Top 20 Holmes-Rahe Stressors	159
6.21	Forty stress categories from our activity ontology with a direct relation to Figure 6.20.	160
6.22	Histogram showing the number of users associated counts of stress related queries.	162
6.23	Listing of the top twenty stress related categories (and associated counts) within the AOL data set.	163
6.24	CUSUM chart of the number of <i>Anxiety</i> related queries over time. Anomalous counts are shown on days fifty nine and sixty six.	164
6.25	Top ten listing of the number of <i>Anxiety</i> related queries made per user. . . .	164
6.26	Top 10 anxiety related query terms for user 1 of Figure 6.25	165
6.27	Breakout of the distribution of activities associated with each behavior associated with our “selective mutism” user.	165
6.28	Stacked graph of distress related behaviors over time.	166
6.29	Two month plot of the number of URLs visited per day for our hyperstress candidate.	169
6.30	User 1 personal (blue) and work (green) behaviors extracted using LDA. . .	169
6.31	Plot of User 1’s work related browsing activities.	170
6.32	CUSUM chart of User 1’s work related activities.	170
6.33	CUSUM Chart of User 1’s personal activities.	172
6.34	Historical plot of Google Trend data for the query term “taxes”.	173
6.35	Plot of Google Trend data for the term “taxes” for the year 2006.	173

6.36	Plot of daily query and session counts for queries related to our “tax profile”.	174
6.37	CUSUM chart of sessions/day for all queries related to our “tax profile” with significant anomalous activity detected on 15 through 17 April.	175
6.38	Mapping of 150 insider threat cases to behaviors violating criteria outlined in the Adjudicative Guidelines for Determining Eligibility for Access to Classified Information.	177
6.39	Cluster one consisting of persistent gamblers. Red squares represent a user having a positive correlation to the activity (darker the color, the more positive the correlation). The <i>Games</i> category consists almost entirely (92%) of <i>Games/Gambling</i> related queries.	179

Chapter 1

Introduction

In 2005, the Air Force Chief of Staff General Michael Moseley announced the new Air Force mission ending with the statement “to fly and fight in Air, Space and Cyberspace”. Just over five years later, the Air Force as well as the rest of the Department of Defense, is still struggling to identify what components are needed for offensive and defensive cyber operations to be viable options for a combatant commander.

The DoD’s present dominance in the kinetic world stems from significant time and effort spent outlining what information is required for planning operations, developing a comprehensive language to unambiguously communicate and share this information, and creating advanced algorithms and mathematical techniques to aid in kinetic planning and decision making. Although the computers and technology forming the basis of cyber operations have been in place for a number of years, compared to their kinetic counterparts, cyber operations are still relatively immature. Figure 1.1 is a table depicting kinetic and non-kinetic characteristics, highlighting the relative immaturity of cyber operations compared to their kinetic counterparts. While mission planning science and technology in the kinetic domain,

<i>Type of Warfare</i>	<i>Age of History (years)</i>	<i>Age of Hardware (years)</i>	<i>Speed of Delivery</i>
Infantry	2,000+	30+	3 - 5 mph
Maritime	1,000+	20+	20 - 40 mph
Armor	90+	20+	30 - 50 mph
Aviation	80+	20+	300 – 1,500 mph
Cyber	10+	18 - 36 mos	186, 282 mps

Figure 1.1: Table depicting kinetic and non-kinetic characteristics, highlighting the relative immaturity of cyber operations compared to their kinetic counterparts.

such as weapon-target assignment and operations research-based algorithms that support the creation of Air Tasking Orders (ATO) are mature and well exercised, the analogous science and technology for computer network operations (CNO) and integrated kinetic and non-kinetic planning are essentially non-existent. Not only are the corresponding basic ingredients missing from the analogous CNO domain, but the fundamental research required to adequately understand and address the issues is unavailable. Figure 1.2 is a quad chart representing what we believe to be the core areas needed for effective cyber operations to take place.

The first quad of the chart represents the foundational information and knowledge required for all cyber operations. It includes a cyber ontology, a weaponization process, well

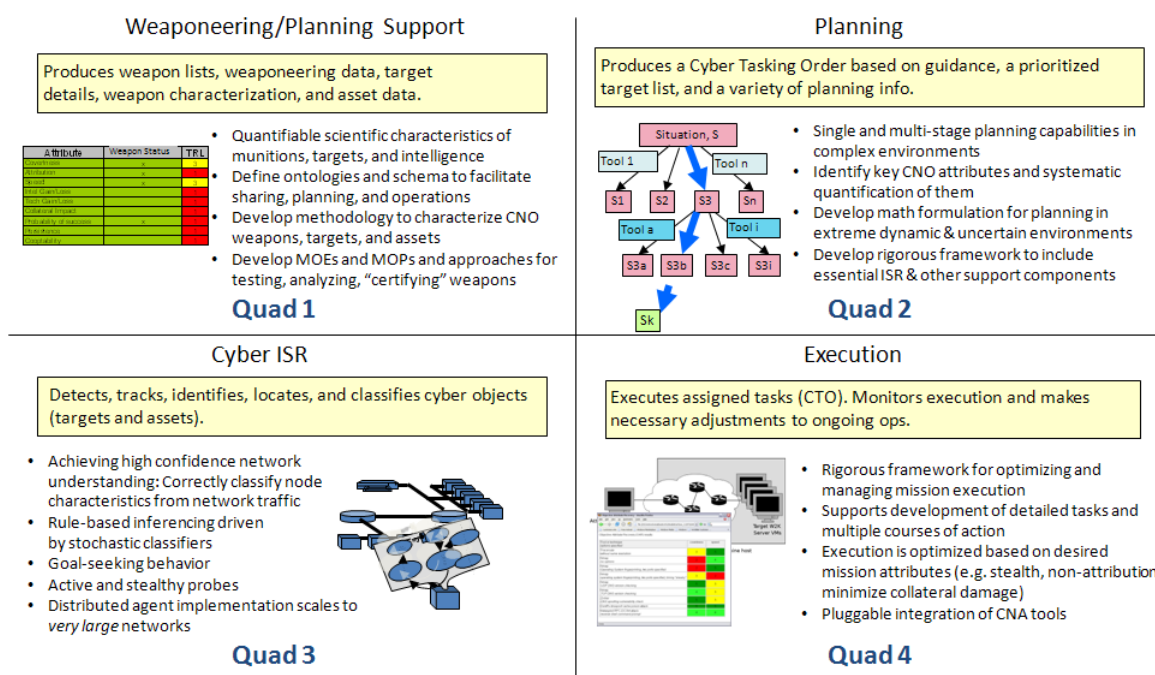


Figure 1.2: Quad chart representing the core areas needed for effective cyber operations.

defined metrics and a methodology to test, characterize, and measures effectiveness and performance of a given tactic, technique or procedure. The fourth quad of the chart deals with the operational execution of cyber missions and the overhead and maintenance associated with them. While increased attention is being given [123][166] to developing the tools and techniques to execute offensive and defensive cyber operations in support of the first and fourth quadrants of the chart, very little emphasis or progress has occurred in planning (second quad) or in assessing the associated information required (third quad) to execute CNO. Although planning is of critical importance in conducting cyber operations, as stated in Joint doctrine [58], before military activities in the information environment can be accurately and effectively planned, the “state” of the environment must be understood. We refer to this state of the environment as cyber situational awareness.

1.1 Cyber Situational Awareness

While long considered an important aspect of strategic and theater planning, situational awareness (SA) is the linchpin to both cyber planning and execution. At its core, cyber situational awareness encompasses understanding the environment in terms of how information, events, and actions will impact goals and objectives, both now and in the near future. Cyber SA, like its kinetic counterpart, deals with complex, dynamic interdependencies that evolve in ways not always well known or understood. Joint Information Operations (IO) doctrine [58] defines three layers of information inherent to the information environment; *physical*, *informational*, and *cognitive*. These layers, often referred to in the kinetic realm as Modified Combined Obstacle Overlay (MCOO) ¹, are the foundation of cyber situational awareness.

The *physical* layer of the information environment includes the “people, place, things, and adversary information capabilities” and may include geographic coordinates of infrastructure, identification of critical links and nodes, and types/quantities of information infrastructure. While collecting, analyzing, and interpreting this information is by no means a solved problem, a great deal of research and resources [231] already focus in this area. Figures 1.3 and 1.4 graphically build out the geographic and logical aspects of the physical layer of the cyber SA model. Figure 1.3 represents the foundation of the model, the association of the “people, places, and things” of interest to points on a map, while Figure 1.4 further builds upon this, overlaying the communications infrastructure on top of the physical layer.

The *informational* layer of the environment focuses on the systems and networks, where “information is created, processed, manipulated, transmitted, and shared”. While doctrine

¹According to the DoD Dictionary of Military Terms [60], MCOO is defined as “a joint intelligence preparation of the battlespace product used to portray the effects of each battlespace dimension on military operations. It normally depicts militarily significant aspects of the battlespace environment, such as obstacles restricting military movement, key geography, and military objectives”.



Figure 1.3: Depiction of the geographic aspects of the cyber situational awareness model's *physical* layer.

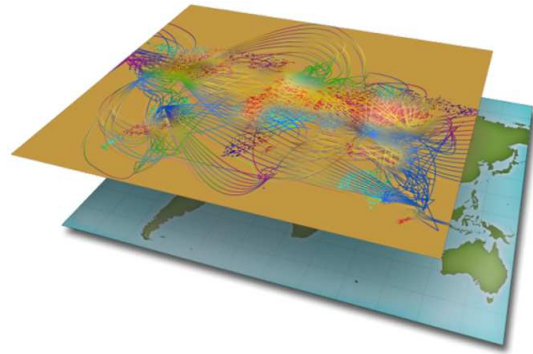


Figure 1.4: Depiction of the communications and infrastructure aspects of the cyber situational awareness model's *physical* Layer

lumps aspects of social interaction into this layer, this research is only concerned with the functional characterization of the systems and networks. We define this as the logical layer of cyber situational awareness. Figure 1.5 is an updated view of our cyber SA model with a visual representation of this layer.

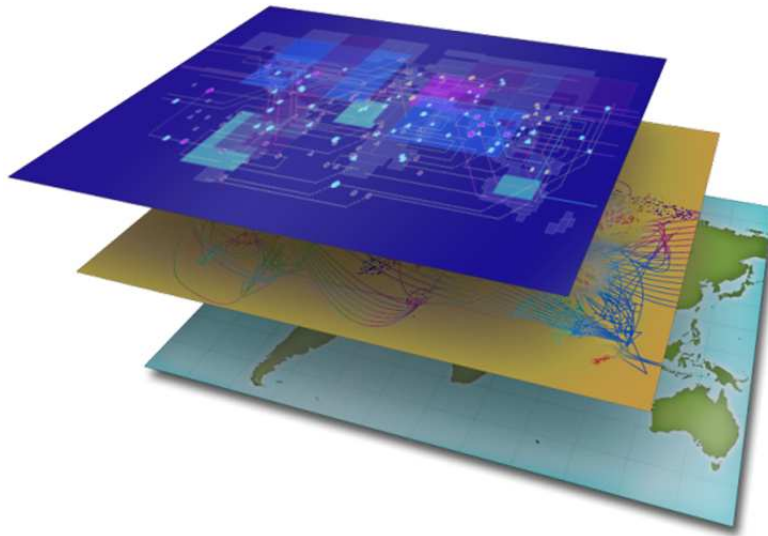


Figure 1.5: Updated view of the cyber situational awareness model with the *informational* layer added.

This figure represents the individual systems and networks connected together via the phys-

ical layer. A large number of tactics, techniques, tools and procedures currently exist for collecting information about and characterizing a specific technology or technology system. Network defense, penetration testing, and the hacking community at large provide both open source [146] [152] and commercial [91] [52] tools for gaining information about various types of machines and technologies. In addition, numerous texts are written outlining the steps to efficiently and effectively profile an individual's or organization's computer system [186][67][220]. Though not a solved problem space, this area is relatively mature with a great many tools and techniques already in place.

Finally, the *cognitive* layer of the environment represents the psychological, cultural, behavioral, and other attributes influencing information flow and interpretation of data by individuals or groups. This may include basic demographic type information such as language, education, religion, etc., but usually relates more to the identification of key individuals or groups and the ability to create profiles to predict how they will perceive, plan, and act given various stimuli. The Army recognized the benefit in this type of situational awareness information and in 2006 stood up the Armies Human Terrain System (HTS) [16]. The HTS provides commanders and staff with information for a given population based on the observations and analysis of social scientists in the field. This information is used by commanders to aid in the interpretation of cultural and regional behaviors and to determine what effect military actions may have on them. While this been fairly successful for the Army, little to no effort has been made to extend this approach into the cyber arena. This research expands the human terrain concept into the cyber realm, by creating a Human Terrain of Cyber Space, thus providing commanders and decision makers a more comprehensive and real time view of the environment in which they are operating.

1.1.1 Cyber Behaviors

Technology today plays a huge role in how many societies function on a day to day basis. Cell phones, personal computers, laptops, and personal digital assistants represent a small number of the technology-based devices used around the world to communicate, work, shop, and play. As more time is spent using these devices, increased amounts of information about the individual using them is being communicated in an online manner. E-mails, purchases, and search queries all leave potentially identifiable traces of who we are. E-commerce and marketing firms started taking advantage of this information years ago, obtaining details on individual's purchase history (online and offline), finance records, magazine subscriptions, supermarket savings cards, surveys, and sweepstakes entries to create profiles they can analyze and manipulate to their needs [4][114][193][175].

Shopping, however is not the only source of data for generating an online profile. Both large government agencies as well as grassroots, informal community groups tend to codify and standardize organizational roles and processes through some combination of training, doctrine and adaptation. In today's technology charged workplace, individuals play these roles and collaboratively execute business processes via web browsers, e-mail clients, and file shares. These activities all leave a cyber fingerprint which can then be analyzed, interpreted, and recreated. Figures 1.6 and 1.7 represent the final layers of the cyber SA model and portray how the individual and the technology interact to create and define one's cyber behaviors. While shown as two separate layers, it is important to note cyber behaviors only exist when both layers are present. The collection, identification, and analysis of these cyber behavioral layers represent the primary focus of this research.

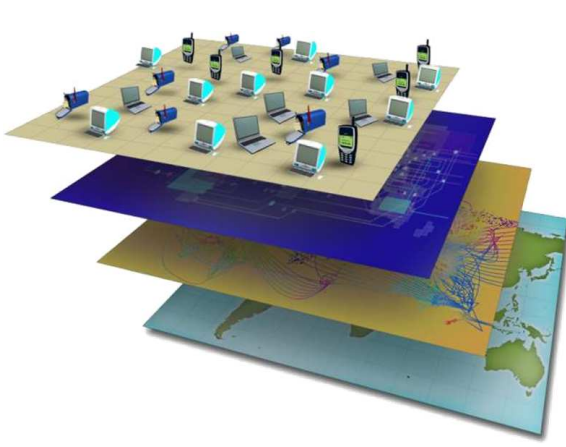


Figure 1.6: Graphical depiction of the mediums used to interact with the cyber environment.

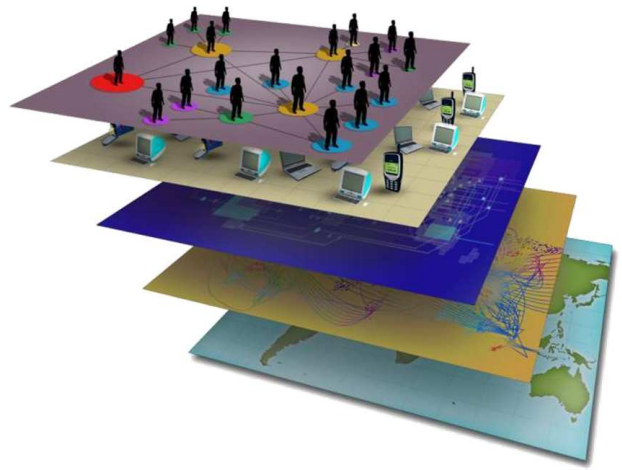


Figure 1.7: Graphical depiction of the interactions and groupings which occur in the cyber environment.

1.2 Behavioral Modeling Framework

The first step in attempting to represent and analyze cyber behaviors is developing a framework to accurately model them. We begin this process by first describing what behavior is. While many definitions exist, we characterize behavior in terms of “observable actions”. Leveraging previous work in this area [189], we define *cyber behavior* as a set of *activities* together with a statistical characterization of those activities. While this may seem fairly straight forward and concise, trying to instantiate and model an individual’s cyber behavior based on this concept is a non trivial matter.

Activities in a cyber context can range from the broad (i.e. surfing the web) to fairly specific (i.e. querying “human behavior”). A user may take part in a number of fairly complex activities, which when combined under certain conditions, describe one or more distinct behaviors. For example, a user browsing web sites on cars, financing, and car dealerships could be characterized by the activity “web browsing”. This “top level” activity however, does nothing to capture the underlying cyber behavior associated with buying a

car.

From our definition, *activities* are obviously a key component in the characterization of ones behavior, but as just demonstrated, *activities* themselves can often be expressed in varying levels of abstraction. In order to describe activities in this manner, we make use of a hierarchical *activity tree*. The root node represents the most basic instantiation of the activity, while the leaf nodes provide further detail and delineation into the specific act. Figure 1.8 is a graphical depiction of a portion of the *Shopping* activity tree for a user. Each sibling leaf

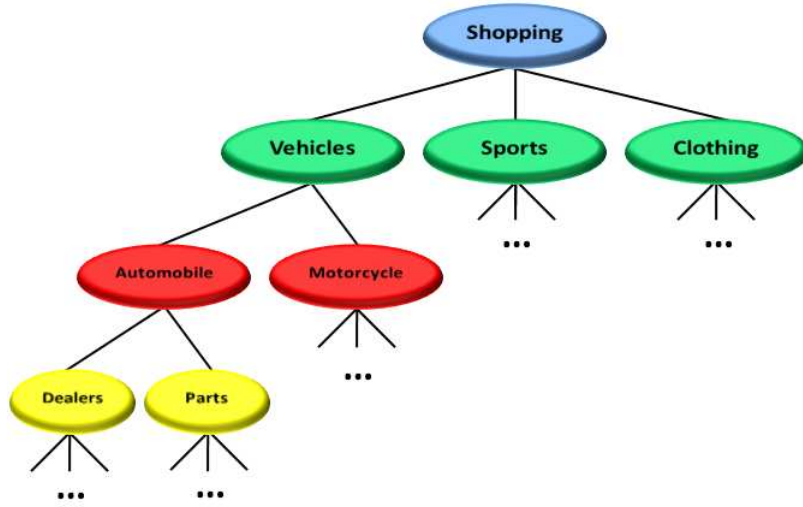


Figure 1.8: Hierarchical activity tree for *Shopping*. Sibling nodes in the diagram are not in and of themselves independent activities, but rather a more detailed description of the *root* activity (i.e. *Automobiles* is not an activity, but *Shopping/Vehicles/Automobiles* is).

node is not in and of itself an independent activity, but rather a more detailed description of its *root* activity (i.e. *Automobiles* is not an activity, but *Shopping/Vehicles/Automobiles* is). In this figure, *Shopping* may be an adequate description of the activity given the context of the analysis. However, for other research (i.e. e-commerce), a more detailed differentiation as to the specifics of what the user is shopping for is of critical importance. Using this

hierarchical activity tree approach, we allow for varying levels of activity representation.

While Figure 1.8 depicts only a portion of an activity tree, in general these structures may consist of hundreds of nodes and be multiple layers deep. Even an activity as seemingly straightforward as *Shopping* quickly becomes extensive when trying to capture the depth and breadth of shopping related activities a user performs. To complicate matters further, activities can be interpreted in multiple ways depending on the behavior being exhibited. For example, a user shopping for baby clothes may be looking for a gift for a new or expectant mother, whereas someone shopping for baby clothes, cribs, diapers, car seats, etc., may be expecting a child. These behavioral semantics are captured through interactions with the environment, activities within the same tree, and activities within other trees.

Due to the complex nature of what an activity tree represents, we view these structures as *activity systems*. We define *system* as a functionally, physically, and/or behaviorally related group of regularly interacting or interdependent elements which taken together form a unified whole [60][61]. Using our shopping example from Figure 1.8, the leaf nodes represent the related group of interacting and interdependent elements which form the unified whole of the activity. These activity systems have a well defined hierarchical structure, characterized by a root node and multiple layers of sibling nodes. In addition, each system has temporal characteristics associated with individual sibling nodes or the root node capturing the “when” and “how often” information associated with an activity. This data may come from browser history files, web server logs, or any other timestamped cyber observable. Figure 1.9 shows a number of independent activity systems a user may have. Each large colored oval represents the system (*Shopping*, *Hobbies*, *Work*, etc.) associated with a particular activity, while the overlaid line diagram denotes the temporal aspects of each lower level node in the activity tree (i.e. how often the user shops for cars, clothes, etc.).

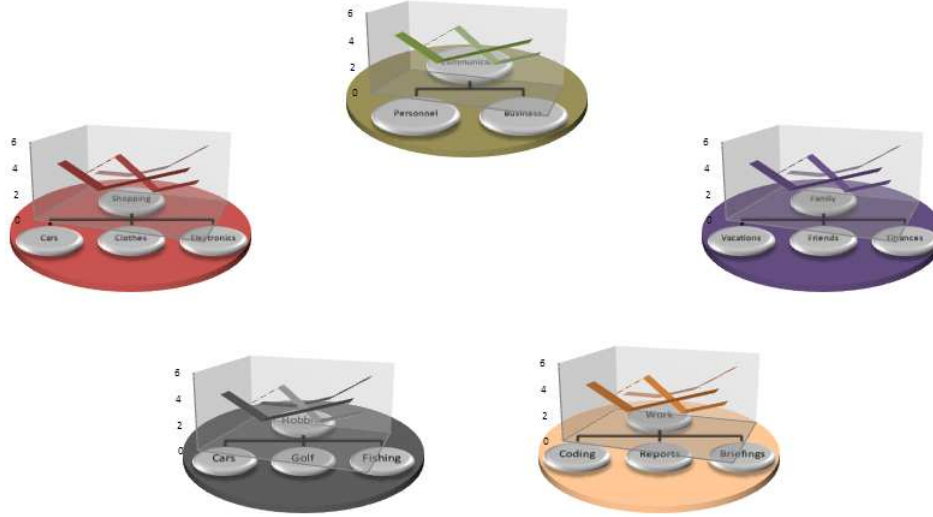


Figure 1.9: Independent activity systems for a user. Each large colored oval represents the system (*Shopping*, *Hobbies*, *Work*, etc.) associated with an activity, while the overlaid line diagram denotes the temporal aspects of each lower level node (i.e. how often the user shops for cars, clothes, etc.).

Given a representation for one's activities, we now address the behavioral context in which these activities are performed. For example, someone's behavior at work is much different than their behavior at home. Their work behaviors in turn differ depending on whether they are dealing with peers, superiors, or subordinates. We define this *behavioral context* as the *environment*. The environment is the physical location or surroundings (i.e. such as being at work versus being home) combined with the mental state (i.e. woke up in a bad mood) of the individual. We make the assumption this behavioral environment and our interaction with it is the determining factor for all of our behaviors and directly affects our activities in both the cyber and non-cyber realm. Generally, individual activity systems alone are unable to adequately represent behaviors unless a very focused behavior is involved (i.e. a gambling addict may only have gambling related activities). It is not until we show the interactions between these systems that we begin to extract behavioral information.

To take these interactions into account, we represent a user's cyber activities as a system of systems (SoS). A SoS is a set or arrangement of systems resulting when independent and useful systems are integrated into a larger system to deliver unique capabilities [154]. Figure 1.10 is a graphical summary of our system of systems representation of a user. The large

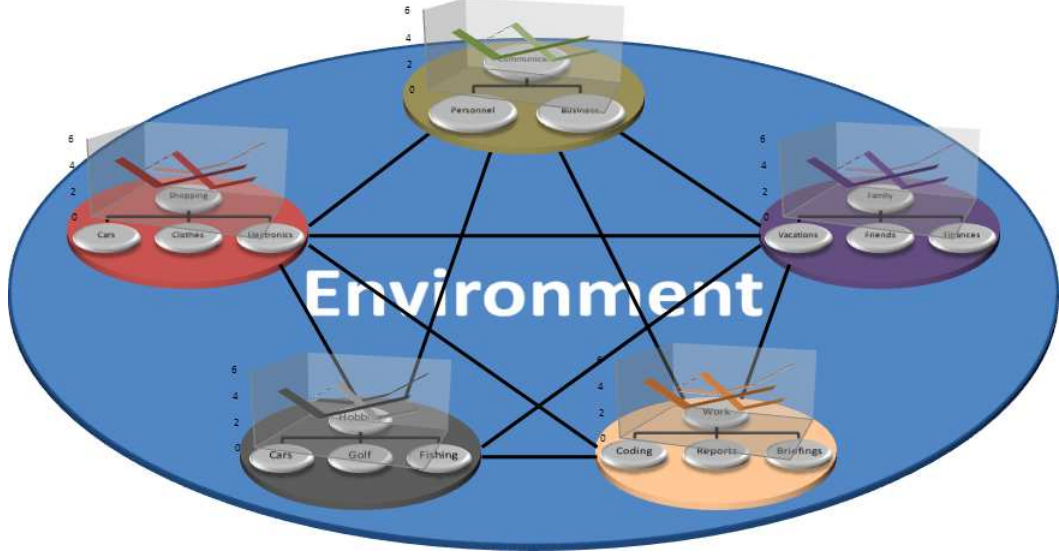


Figure 1.10: System of Systems representation showing all possible interactions among activity systems of a user. The large blue oval represents the behavioral environment.

blue oval represents the behavioral environment. The activity systems are overlaid on this to symbolize their interaction and dependence with the environment. The lines connecting the activity systems depict all *possible* interactions taking place among systems. A particular *behavior* is then just an instantiation of a subset of these activity systems within this SoS construct. Figure 1.11 is a graphical instantiation of this concept using our car buying reference made earlier. In this diagram, the lines between lower level activities represent the linkages (these may be Markov based, correlation based, etc.) between the specific activities within the individual activity systems while the line chart is a temporal representation of

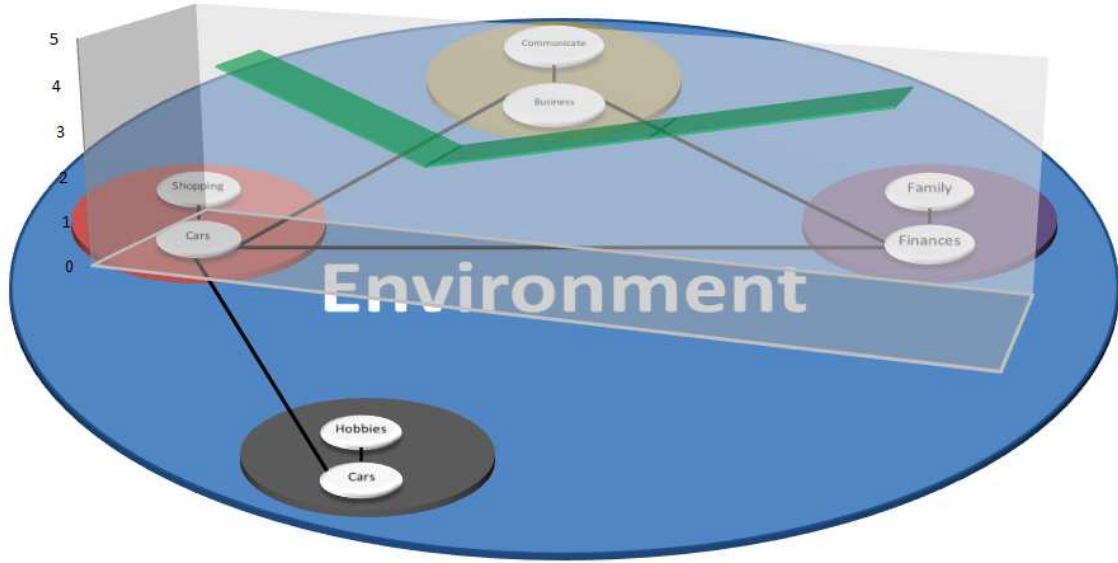


Figure 1.11: User’s “car buying” behavior extracted from the SoS diagram of Figure 1.10.

the behavior over time. Looking at the user as a whole (Figure 1.10) or at the individual activity systems (Figure 1.9), it is impossible to extract this behavioral information. It is not until we examine the underlying structure in and among these activity systems that we can infer behavior in a quantitative manner. We instantiate our SoS behavioral model in this dissertation, and provide the algorithms required to extract these behaviors.

With our behavioral model described, we now briefly outline our behavioral modeling methodology as depicted in Figure 1.12. The first step is to collect behavioral information from subject’s online (Internet based) and/or offline (non-Internet based) computer activities. The data is then pre-processed to minimize irrelevant information and noise. The Activity Discovery Engine (ADE) “normalizes” the data by mapping it to a hierarchical ontology allowing for the population of activity trees. Individual and group activity data are then extracted and stored in the Behavioral Activity Database (BAD) for further analysis. Finally, advanced analysis techniques are used on the BAD data in areas such as behavioral characterization, prediction, and anomaly detection. The remainder of this dissertation de-

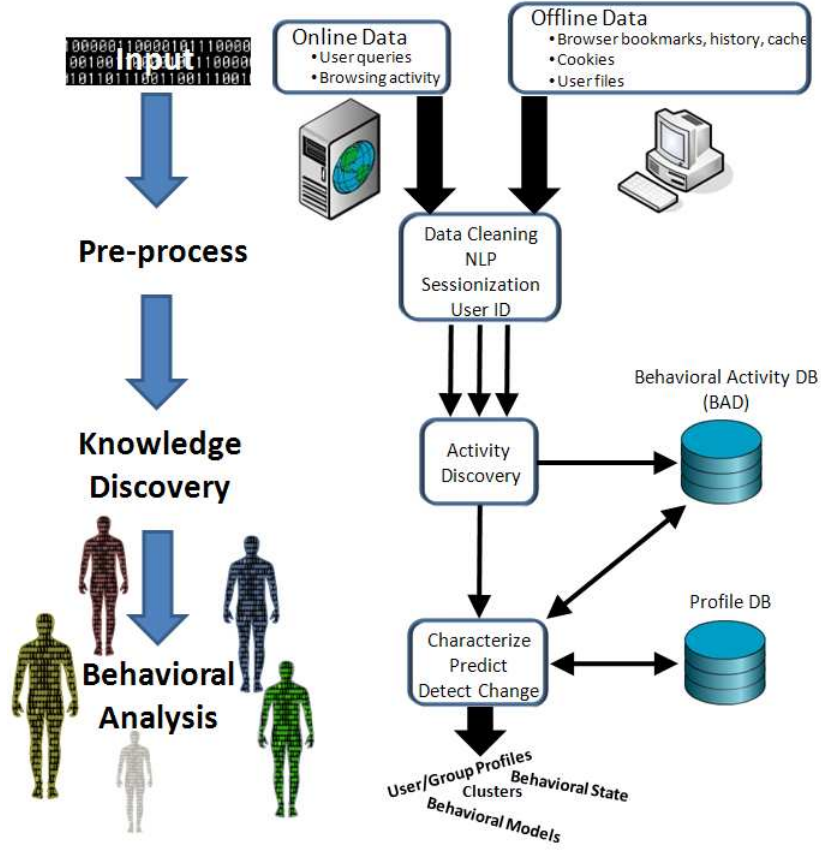


Figure 1.12: Graphical representation of our behavioral modeling methodology.

lineates each step of the methodology.

1.3 Contribution of Work

This thesis embodies several contributions.

1. We have implemented a novel approach to the modeling and analysis of behaviors based on a user's cyber characteristics. By formalizing and instantiating our behavioral model defined in Section 1.2 on user data, we have in place a mechanism to characterize and profile both individuals and groups at varying levels of fidelity. We are currently unaware of any such mechanism in existence in the cyber realm at this time.

2. We develop and implement a mechanism to extract and normalize cyber activities from a wide range of data sources. Previous work in this area has primarily focused on individual data types and does not address the depth or breadth of analysis presented in this work. We believe these data fusion techniques may open up research in the area of semantic web development.
3. We have implemented an iterative behavioral sample size algorithm based on Bayesian bootstrapping techniques. Most traditional sample size algorithms are based on a “one size fits all” approach and do not allow the flexibility needed to model individual users. Because no two users are alike, the algorithm dynamically determines when enough information has been collected based on Bayesian credible intervals. Leveraging work done in control theory, this sample size is constantly monitored and updated as a user’s behavior changes over time. We believe this work is unique from both an engineering and psychosocial standpoint and is foundational to our methodology.
4. We have employed a new method to identify and characterize transient and persistent behavioral states. The ability to classify behavioral persistence versus transience is a significant achievement in this area and allows for highly accurate behavioral prediction to be achieved.

While we have focused on how this research may be best utilized within the Department of Defense, we believe this work offers great potential in other domains as well. The ability to characterize, predict, and detect change in user behaviors based on their cyber activities is something sought after for a number of reasons. The financial sector may leverage this work for fraud prevention and credit scoring while human resources can track employees by monitoring their behaviors for malicious activity (insider threat) and for classifying and ranking

user’s skills. This work offers a means for e-commerce to expand the scope of its current profiling techniques to better characterize and predict purchase patterns for individuals and groups of online shoppers.

Within the Department of Defense, this work provides the foundation to accurately and effectively represent and analyze the behavioral layers of the cyber situational awareness environment, which in turn will provide planners and decision makers critical information to achieve their mission. This research will provide new insights and analysis methods as of yet unimagined in the planning of offensive and defensive cyber missions. Analysts will no longer be constrained to functional characterization of computer systems using terms such as “client”, “web server”, and “database server”. The capacity to “see” the user at the keyboard will allow for advanced courses of action to be developed and the full power of cyber effects to be achieved.

1.4 Structure of Thesis

The remainder of this thesis is organized as follows.

- Chapter 2 – Provides background on work related to the area of behavioral modeling.
- Chapter 3 – Outlines the types cyber observables which are addressed by this work and the pre-processing steps needed before labeling can be performed.
- Chapter 4 – Describes our activity ontology and details the algorithms used to transform cyber inputs into activity labels. The chapter concludes with an empirical evaluation of this algorithm.

- Chapter 5 – Describes the application of mathematically derived principles to cyber-based activities for the purposes of characterization, prediction, and change detection in user and group behaviors. In addition, we implement our behavioral model and outline the methods used to analyze and interpret this model in a quantitative manner.
- Chapter 6 – Provides an empirical evaluation of the benefits of using cyber-based behavioral modeling in three disparate domains; military targeting, stress monitoring, and insider threat detection.
- Chapter 7 – Outlines areas of interest for future research and development.

Chapter 2

Previous Research on Cyber-Behavior Analysis

This chapter focuses on the major influences at work in behavioral modeling and characterization. While literature relating specifically to this area is severely limited, a great deal of supporting research has been conducted. This research can be broken down into four main groups; web usage mining, engineering, government/industry, and psychology. The remainder of this chapter will examine pertinent research in these four areas.

2.1 Web Usage Mining

The area with perhaps the most significant overlap with behavioral modeling is that of web mining. Web mining is a focused area of data mining dealing with the discovery of knowledge from web data. In general, web mining is broken out into three core areas; web structure mining, web content mining, and web usage mining [120]. Web structure mining involves discovering information from the inherent structure of the web and is primarily focused on

hyperlinks. The internal link structure of one specific site or the external link structure between many different sites can yield information relating to sites of interest/importance or may be used to discover user groups or communities. Examples of research done in this area can be seen in [209][77][44][171].

In contrast, web content mining addresses the analysis of the actual web page content. Borrowing on a number of natural language processing techniques, web content mining can be utilized in areas such as e-commerce in order to mine forums and customer review sites to determine general trends in users' beliefs and interests. Projects in this area include [205][163][45][165]. One use of this technology was implemented in the Dark Web research [46][176][177]. The focus of the project was to study and understand international terrorism (specifically Jihadist) through the collection and analysis of web content generated by international terrorist groups, including web sites, forums, chat rooms, blogs, social networking sites, videos, virtual world, etc. Content categories such as recruiting, training, propaganda, etc. were then identified for further analysis. While web content mining offers a means to begin to "paint a picture" of areas of interest for individuals, most current research fails to address this aspect.

The last area of web mining, which is by far the biggest growth area, is web usage mining. Unlike the two previously mentioned aspects of web mining that are primarily concerned with web sites and web data, web usage mining is focused more on the actual end user. The focus of web usage mining is to extract user access patterns by analyzing and interpreting "visit information" from users' browse patterns, also referred to as "click-streams". Clickstream analysis is one of the fundamental analysis techniques used in web usage mining. A Clickstream can best be described as a historical description of what a computer user did and where a computer user went (or clicked on) while browsing a particular

web site. The critical piece of information in a clickstream comes in the form of the Uniform Resource Identifier (URI). The URI represents the global address of documents and resources present on the World Wide Web. The most common form of the URI is a web page address. When a user operates a web browser and requests a URI, or clicks a hyperlink on an existing page, a HTTP/GET request is generated. This GET request can be passively intercepted, pulled from log files, or captured by agent devices installed on the user's machine. Along with the URI, clickstreams will normally contain timestamps of the GET request, the source of the request, the destination host, and assorted browser information.

Web usage mining research has been conducted in a number of fields to include web page pre-fetch/cache, web site optimization, and recommender systems, however, the field generating the most interest (and revenue) is without a doubt e-commerce. Cyber Monday (the online version of Black Friday) sales this year topped \$881 million dollars [117]. That number tops Black Friday's take by over \$77 million and is a fifteen percent increase over last year. With this trend in online shopping turning into the norm, interest in profiling and analyzing online shoppers is at an all time high.

Retail and marketing firms are taking advantage of user profiling by compiling volumes of information on individuals to better target their sales. Such profiling is accomplished by aggregating data on individuals purchase history (online and offline), finance records, magazine subscriptions, supermarket savings cards, surveys, and sweepstakes entries, just to name a few. This information is then cleaned, organized, and analyzed using a number of statistical and data mining techniques to create a basic profile (in this case with a focus on shopping) of that individual. These profiles are then used to better target ad campaigns, personalize a shopping experience, or make recommendations of additional product purchases based on a user's historical interests.

2.1.1 Data Collection/Pre-Processing

Web usage mining data is gathered from a fairly limited number of sources. These sources are primarily divided into two categories; server-side data and client-side data.

2.1.1.1 Server-side Data

Server-side data exists in the form of web server logs and is generated automatically by web server applications. While web server implementations can vary greatly, the logs they generate are very similar. Logs typically exist as flat text files, the most widely used formats being the Common Log File format [221] and the Extended Log format [222]. A standard web log (normally referred to as *access log*) contains fields such as the IP address from where the user request originated, the date/time of the request, the HTTP method of the request (GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, or CONNECT), a numerical status code indicating the response from the server (usually indicating some level of success or failure), and the size (in bytes) of the transaction. Additional log files may include the *referrer log*, providing information about the web page where the request originated (i.e. if a user was redirected from another page), and the *agent log* which will indicate the type of browser used in making the request. Depending on the server implementation, aspects of these individual files may be combined into one. An example of a typical log entry from an Apache web server is shown below.

```
10.10.10.101 - - [10/Dec/2008:12:08:33 -0500] "GET / HTTP/1.1" 200 477
"http://www.dartmouth.edu" "Mozilla/5.0 (Windows; U; Windows NT 5.1;
en-US; rv:1.9.0.4) Gecko/2008102920 Firefox/3.0.4"
```

This log entry shows the request coming from the IP address 10.10.10.101 on 10 December and is a request to the root directory at `http://www.dartmouth.edu`. The web browser used

is Mozilla Firefox and the machine the request was made from was Windows XP. While this request shows the incredible detail of the information included in server side logs, it also illuminates a significant shortcoming in using this type of data. The IP address 10.10.10.101 is a private IP address and therefore not Internet routable. An address of this type comes from intranets utilizing Network Address Translation (NAT). NAT is a technique that hides an entire address space, usually consisting of private network addresses (RFC 1918), behind a single IP address in another, often public address space [212]. In addition to NATing, caching, multiple users utilizing the same computer, and proxies all can mask server-side data.

Local caching has two functions; to improve browser performance and to minimize network traffic by storing (caching) a copy of the page locally. When a user requests a page that has been cached, no request is ever made to the server and thus no log of the request is ever generated. Proxy servers act in a similar manner but at a mid point in the network (normally at a gateway type position in the network). Many users may utilize the same computer throughout a day making it difficult to extract “individual” data streams when all the information originates from the same IP address. Proxy servers and other gateway type devices, can “mask” the identity of the individual computer user, making the requests all appear to come from the same IP address. Figure 2.1 is a graphical summary of the impediments to using server-side data [20]. While these problems have been addressed and researched in a number of papers and books [43][167][20][120], no true resolution exists at this time.

One mechanism of benefit in the area of user identification is the ‘cookie’. A cookie is a small text file serving as a marker to tag and track user’s browsing activity on individual web sites. The file is stored locally on the user’s machine and can be used to uniquely identify

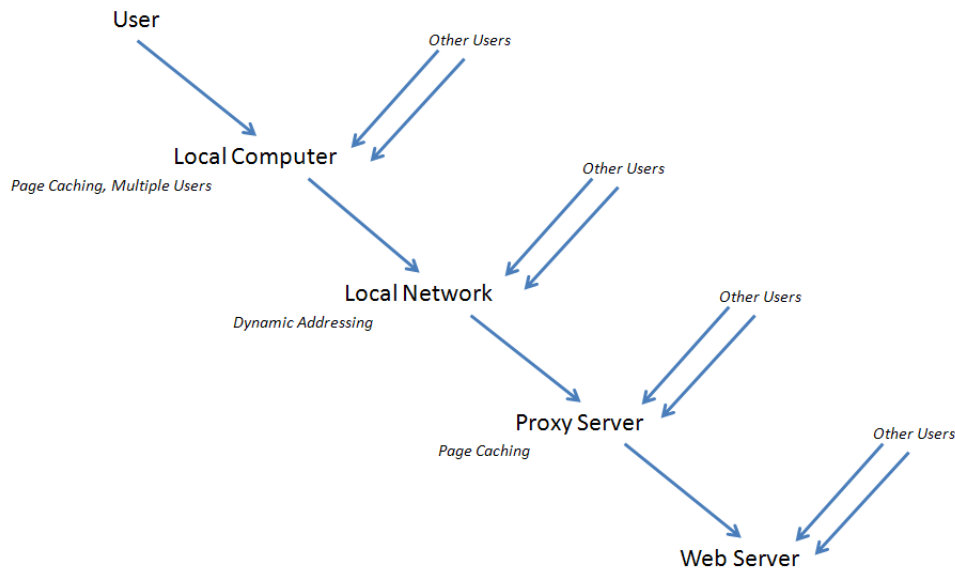


Figure 2.1: Graphical summary of the impediments to using server-side data.

requests to a given site. When the user returns to the same site, the information in the cookie can be accessed by the web server and will identify the user. This mechanism works for multiple individuals using the same machine given each individual has their own unique login (cookies are associated and stored with operating system generated user profiles). Of course cookies only work if they are enabled and the majority of web browsers today will prompt the user to disallow the use of cookies as they are a potential security and privacy threat.

2.1.1.2 Client-side Data

Client-side data collection overcomes a number of the shortcomings outlined by server-side data. The data itself is very similar to the web log data previously displayed, but because there are no standard applications used for collecting this information, there is no standard or well established format for what specifically is being collected. Collection is normally

accomplished through some type of “agent” application running on the user’s local computer. Captured data may be stored locally or dynamically streamed to a centralized storage device. Data aggregation services such as comScore [50] and Nielsen [148] obtain user consent to install client-side applications on user machines to monitor and track browsing activity and patterns.

Advantages to client-side data include much more reliable identification of users as well as direct recording of all browser actions (local or remote caching is not a factor). Due to the accuracy and reliability of this type of data, it is the preferred data capture source for web usage mining.

2.1.2 Analysis

From an analysis standpoint, most web usage mining is focused in three areas of interest; clustering, association mining, and prediction.

2.1.2.1 Cluster Analysis

Clustering is considered one of the most commonly used analysis techniques in web usage mining [120]. User clustering allows the definition and grouping of users exhibiting similar browsing patterns. Clustering based on content [93][227], frequency [121][119], and navigation patterns [82][21] have all been examined in trying to determine the best mechanism to group users exhibiting similar behaviors. The results of these clusterings can then be used in areas ranging from determining the most popular path through a given site to how to best organize a site for a given user base.

Clustering of web pages themselves for the purposes of organization, classification, and search is also an active area of research in Web Usage Mining [211][181][233]. Work in this

area varies greatly and touches on a number of fields to include natural language processing, information retrieval, and artificial intelligence.

2.1.2.2 Association Analysis

Frequent item set mining and association rule induction, also referred to as market basket analysis, are methods used to find regularities in the shopping behavior of customers [4][5]. This technique is used everywhere from supermarkets to e-commerce to find sets of products frequently bought together. With this analysis, one can infer that if certain products are in a “shopping cart” (either literally or virtually), then with a high probability, certain other products should be present. This information is expressed in the form of rules and used in commerce-focused organizations to attempt to increase sales by arranging or presenting items together that match these rules. To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest are typically applied to the item set. The best-known constraints are minimum thresholds on *support* and *confidence*.

Let S be an item set and T the bag/multiset of all transactions under consideration. Then the *absolute support* (or simply the *support*) of the item set S is the number of transactions in T that contain S . Likewise, the *relative support* of S is the fraction (or percentage) of the transactions in T which contain S . *Support* is a useful measure because if it is too low, the rule may just occur due to chance. The support of the rule $X \rightarrow Y$ is computed as follows:

$$support = \frac{(X \cup Y).count}{n}$$

To measure the quality of association rules, Agrawal et al [5] introduced the confidence of a rule. The confidence of a rule, $X \rightarrow Y$, is the percentage of transactions T that contain X also contain Y . It is computed as follows:

$$confidence = \frac{(X \cup Y).count}{X.count}$$

Confidence determines the predictability of the rule. If the confidence of the rule is low, one cannot reliably infer or predict Y from X . A rule with a very low predictability is obviously of limited use.

Apriori, Eclat, and FP-Growth are well known algorithms used to uncover frequent item-sets of interest. Amazon is well known for using this type of information through their “Frequently Bought Together” and “Customers Who Bought This Item Also Bought” user recommendations. A large corpus of research exists in using association analysis for this type of recommender system [235][114][190][115].

2.1.2.3 Prediction

Modeling and predicting a user’s surfing behavior on a web-site is an active area in WUM with applications in areas involving search engine improvement [37], web cache optimization [194][25][159], prediction and influence on buying patterns [47], and personalizing the browsing experience [168].

Markov models are perhaps the most popular strategy used in WUM to analyze and predict behavior [55][191][238]. The basis behind a Markov model is the Markov property. A stochastic process has the Markov property if the conditional probability distribution of future states of the process, given the present state, depends only upon the current state. In other words, the description of the present state fully captures all the information that could influence the future evolution of the process. Given a list of pages a user has accessed over a finite period of time as input, it is a fairly straightforward exercise to compute a first-order transition matrix and from that a Markov model. Research [11][40][55] shows first-order Markov models are relatively unsuccessful in predicting a user’s next action. This is because

these models do not look far enough into the past to correctly discriminate the different behavioral modes of the different users. In order to obtain better predictions, higher-order models are required. A k th-order model is defined so that the prediction of a given state depends on the previous k states in the model. Unfortunately, these higher-order models also have a number of limitations in regards to (i) high state-space complexity, (ii) reduced coverage, and (iii) the potential for worse prediction accuracy. A number of ways have been proposed to get around these problems ranging from pruning the state space of the k th order models [55] to using mixtures of Markov chains [40].

2.2 Engineering

A number of traditional engineering disciplines exist to address various aspects of behavioral modeling and analysis. While a complete review of these disciplines is well beyond the scope of this research, a brief synopsis of each as well as some of the contributing areas will be described in the following sections.

2.2.1 Information Retrieval

Information retrieval (IR) is defined as “finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).” [127]. Unstructured data is defined as any data not having clear, semantically overt structure easily interpretable by a computer. One of the areas receiving the greatest benefit from IR is web search engines. With computer usage around the world reaching epic proportions, many use the terms “information retrieval” and “search” interchangeably. The majority of search engines store their data in an inverted index, one of

the key data structures used in IR. An inverted index is a dictionary of terms constructed for the data in question (i.e. web pages, documents, e-mails, etc.). For each term, there exists a list recording which documents the term occurs in and often the offset location within the document. Given the documents $D_0 = \text{"It is what it is"}$, $D_1 = \text{"What is it"}$, and $D_2 = \text{"It is a banana"}$, the following inverted index would be generated where the numbers in the brackets represent the subscripts of the above documents.

```
a:      [2]
banana: [2]
is:     [0, 1, 2]
it:     [0, 1, 2]
what:   [0, 1]
```

Once an inverted index is created, full text search can be accomplished by jumping to the word IDs (via random access) in the inverted index much faster than using sequential access.

While the tie between IR and behavioral modeling is not completely obvious at first glance, IR storage and retrieval techniques serve as the foundation for a number of classification and clustering engines that may be utilized in this area.

2.2.2 Artificial Intelligence

Artificial Intelligence (AI) is one of the first fields of study to attempt to model human behaviors in a computational manner. While definitions of AI vary greatly from one text to the next, a set of common themes can be extracted [184]; systems that think like humans, systems that think rationally, systems that act like humans, and systems that act rationally. Natural Language Processing (NLP), knowledge representation, automated reasoning, and machine learning are distinct areas within AI addressing the various aspects of these broad themes.

2.2.2.1 Natural Language Processing

Natural language processing (NLP) is a field of computer science concerned with the interactions between computers and human (natural) languages. The ability to interpret and “understand” what a user is reading, writing, or searching for is a key aspect of behavioral modeling. NLP tools and techniques are used extensively in Information Retrieval (Section 2.2.1) and Human-Computer Interaction (Section 2.4.1) in applications ranging from automated help desk systems [107][13] to aiding those with physical handicaps [141]. These techniques are particularly useful in our research for data preparation and cleaning purposes.

2.2.2.2 Knowledge Representation and Reasoning

Knowledge representation is an area in artificial intelligence concerned with how to use a symbol system to represent a given domain of discourse. Using functions that may or may not be within the domain it is then possible to infer and reason about these objects in an algorithmic manner. Within this context, a domain of discourse is an analytic tool used in deductive logic to indicate the relevant set of entities being dealt with by quantifiers. Due to this connection between knowledge representation and reasoning, the two are often discussed together.

One of the first attempts to replicate human reasoning was the belief-desire-intention (BDI) model [36]. The goal of using this type of architecture was to allow an agent to choose its actions and make decisions for reasons similar to those a human decision maker would use. While a bit dated, BDI models are still prevalent in many agent-based systems [89][230].

Many of the recent advances in knowledge representation and reasoning were made thanks to the computer gaming industry. Games such as “The Sims” and “Half-Life” were designed

to replicate human behaviors and reasoning in large scale complex environments.

2.2.2.3 Machine Learning

Machine learning is the subfield of artificial intelligence concerned with the design and development of algorithms allowing computers to improve their performance or learn over time based on data, such as from sensor data or databases. One of the major interest areas in machine learning research is to automatically produce/induce models and/or patterns from data. This type of AI has been used in areas ranging from robotics [15] to computer games [34].

2.3 Government/Industry

Government and industry both have a vested interest in behavioral modeling of individuals, but for vastly different reasons. The ability to identify, track, and predict behaviors of individuals can be applied in areas ranging from capturing terrorists to corporate fraud. The next two sections outline the predominant applications in this area.

2.3.1 Social Network Analysis

Social Network Analysis is an approach to map, measure, and analyze relationships and flows between individuals, groups, and organizations. Tracing its origins to classical sociology in the early 1950's, Social Network Analysis has been applied in organizational and social psychology, sociology, anthropology, sociolinguistics, geography, communication studies, information science, organizational studies, economics, and biology. The last ten years have seen a huge growth in the number and types of networks in which humans interact, resulting

in renewed interest in Social Network Analysis by industry, government, and academia.

One of the first goals of social network analysis is to be able to visualize and analyze relationships between people and/or groups. This is usually accomplished using graph theoretic calculations which study the behavior of networks and multivariate analysis to create visual displays. By focusing on patterns of relations between and among people, organizations, states, etc., social network analysis aims to:

- Describe networks of relations as fully as possible
- Identify prominent patterns in such networks
- Trace the flow of information through them
- Discover what effects these relations and networks have on people and organizations

It is these very characteristics that make social network analysis enticing to law enforcement and the Department of Defense. The ability to track and uncover associations of criminals and terrorist organizations has traditionally been a very time intensive and manually exhausting process. Using data from a variety of sources, social network analysis can be an effective tool to analyze these networks and uncover previously unseen relationships. Figure 2.2 is social network diagram created by Valdis Krebs, demonstrating how media reports alone could be used to create a detailed network of the terrorist organization associated with the September 11 attacks against the World Trade Center [110]. More recently, detailed networks such as these were created to trace the kinship network of Saddam Hussein prior to his capture [24].

The emergence and ever growing popularity of social networking sites such as Friendster, Facebook and MySpace have helped to bring social network analysis back into the mainstream. According to Alexa.com, a web trafficking service, as of January 2009, MySpace

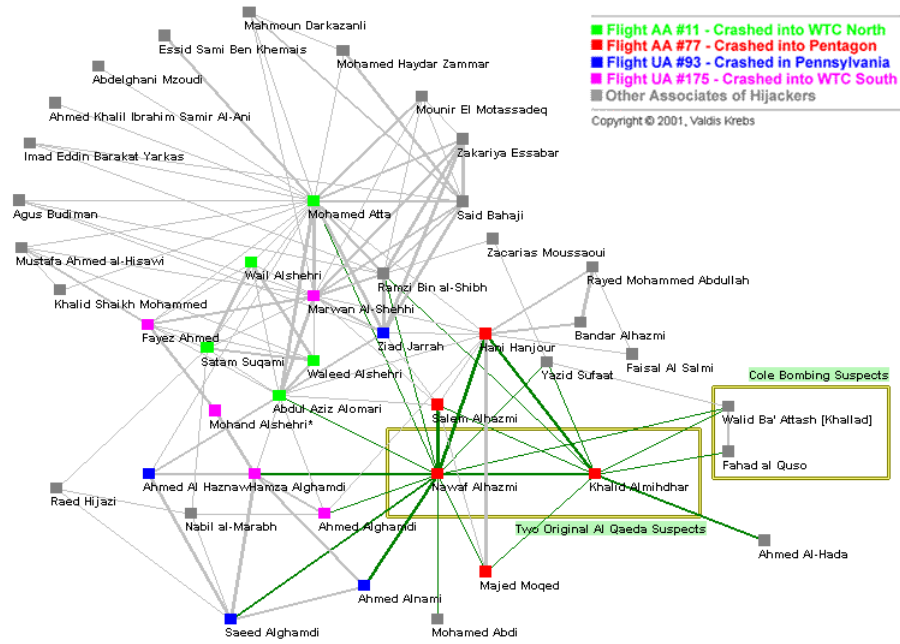


Figure 2.2: Social network diagram demonstrating how media reports can be used to create a detailed network of the terrorist organization associated with the September 11 attacks against the World Trade Center.

and Facebook were the sixth and eighth most popular websites in the world. These sites allow users to build friendship networks, share photos, blogs, user profiles, and messages. A great deal of research and analysis has been done on these sites to evaluate and study the formation, connectivity, and evolution of social groups and networks [19][66][2].

2.3.1.1 SNA Data

Wasserman and Faust describe two types of variables in a social network data set; structural and compositional [229]. Structural variables measure ties between a given pair of actors and form the foundation for SNA. This data is typically portrayed in SNA diagrams as the vertices and lines of a graph. Examples of structural variables include friendships on a social networking site or the interactions in a terrorist hierarchy. Composition variables, on the

otherhand, are more based on the individuals themselves. Examples of this type of variable include age, gender, race, or socioeconomic status.

Traditional SNA techniques employed to collect both structural and compositional variable data include questionnaires, interviews, observations, archive records, and experiments [229]. Increases in the types and uses of technology over the years resulted in the collection of SNA data becoming both significantly easier and significantly harder at the same time. Cell phone records, computer logs, emails, and the vast array of other collaborative data provide a wealth of social network interaction residuals. With proper authority and access, this data is fairly easy to collect and utilize to define and analyze social networks. At the same time, the abundance of data makes SNA more difficult since it is nearly impossible to ensure one has captured all of the relevant interactions for a given group of individuals of interest. Leaving out a communication mechanism such as SMS may result in assumptions being made that could easily be proven false given access to this information.

2.3.1.2 Analysis

While a complete review of SNA techniques is beyond the scope of this paper, the most important and widely used conceptual tools are best described in one word; centrality. While analysis goals may vary, a common theme involves identification of the most important actors within a network. SNA studies typically produce descriptive results such as which actor is the most “central”, which actors belong to which group, and which actors are roughly equivalent to one another.

The shape of a social network also gives insight related to the network’s usefulness to its individuals. A smaller, tighter network may be less useful to its members than networks with lots of loose connections or weak ties to individuals outside the main network. More

open networks, with many weak ties and social connections, are more likely to introduce new ideas and opportunities to their members than closed networks with redundant repeating ties. [197]

While SNA is an effective tool for certain aspects of behavioral modeling, as its name implies, it is primarily focused on the social networks and not on the individuals making up these networks. SNA provides an analyst tools to evaluate the importance of individuals and groups to the network, but offers little insight into the characteristics that define who the individuals are. We believe behavioral analysis and SNA are complementary techniques which should be performed in parallel in order to provide the most comprehensive representation of the environment.

2.3.2 Insider Threat Detection

While the computer and network security industry continues its exponential growth [22] in an effort to keep “bad guys” out, similar focus and attention was not spent monitoring those already “in”. Although no strict definition exists, insider threat can be characterized as someone with legitimate access to an organization’s computers or networks whose actions put the organization or resources at risk [170]. Insiders may range from a naive user whose uninformed actions reveal sensitive data, to a malicious employee who is bent on revenge over a promotion not received. The focus of this review of literature is on the latter type; deliberate actions that put an organization at risk.

With malicious insider activities on the rise [180], interest in insider threat research and technologies is expanding in both industry and academia, however, little progress has been made to create measures to effectively and efficiently detect insider activities. Approaches

being followed at this time, are best categorized as quantitative and qualitative in nature.

The quantitative approach deals mainly with the statistical modeling and analysis of data for the purpose of identifying trends or norms. Once identified, this data can then be used to identify activities falling outside the “norm” warranting further investigation. Most quantitative approaches are either probabilistic [125][126][124] or policy based [42][41]. While of interest, current quantitative approaches provide little insight into the behavioral aspects of individuals and have had little impact in solving the insider threat problem.

The qualitative approach to insider threat is more holistic in nature taking a focus on the characteristics of the individuals who commit these crimes to create profiles. By identifying insider threat behavioral characteristics from previous cases, the goal is to be able to depict similar trends in individuals who currently match or are beginning to match aspects of these profiles. The most recent work in this area comes from two sources; a joint venture between Carnegie Mellon University and the Secret Service [101][68] and the Defense Personnel Security Research Center (PERSERC) [85][201]. While a complete synopsis of the results of these studies is beyond the scope of this review, a summary of significant findings from their work is shown below.

- Allegiance - In cases dealing with insider threat and the Department of Defense, allegiance to a foreign country or cause more than doubled (to 46%) as the primary reason for committing espionage.
- Drugs, Alcohol, and Gambling - Once considered a significant indicator, insider threat cases based on one or more of these factors have significantly declined since 1990.
- Financial Considerations/Gain - Monetary gain has always been a factor in insider threat cases, but it is those with financial difficulty versus those motivated simply by greed who tend to be in these activities.
- Life Events - As with monetary gain, significant life events tend to be a theme in many insider threat cases. In a recent study, 33% of the cases involved the perpetrator

experiencing a serious crisis in their lives during the six to eight months immediately prior to attempting espionage.

Quantitative approaches provide tools to help corporations and agencies both lock down and monitor suspicious activities within their computer networks, while qualitative methods provide a means to identify behavioral trends and profiles based on past insider threat cases. Unfortunately, there is no mechanism to merge these two areas so behavioral profile characteristics identified in qualitative analysis can be monitored for, tracked, and identified using quantitative approaches. We see this dissertation as a mechanism to bridge this gap.

2.4 Psychology

Psychology is an academic and applied discipline of the social sciences focused on the study of behavior and mental processes. Though comprised of a large number of specialties ranging from school counselors to studying the criminally insane, our work is most interested in those disciplines dealing with the study of individual behavioral and personality characteristics. Relevant work in this area can be seen in aspects of personality, cognitive and social psychology.

Psychologists define behavior as anything an organism does, whether it can be observed directly or must be inferred [228]. Behaviors can be conscious, unconscious, overt, covert, voluntary, or involuntary thus making the ability for psychologists to evaluate and analyze them a non-trivial manner. Because of the diversity in how behaviors can be displayed by an individual, a number of techniques exist for psychologists to attempt to accurately collect and characterize this type of information to include tests, surveys, interviews, and direct observation.

Personality is defined as a dynamic and organized set of characteristics possessed by a person that uniquely influences his or her cognitions, motivations, and behaviors in various situations [185]. Personality psychology is the primary branch of psychology dealing with the study of personalities and has a focused area of research dedicated to constructing a coherent picture of a person and their psychological processes [35]. Another area of emphasis in this field is the study of how people are similar to one another. Changes in personality may occur from diet, medicine, life events, or learning, but most theorists believe a person's personality is relatively stable over time. Personality types, originated by Carl Jung, are a psychological classification mechanism used to differentiate types of individuals. Unlike personality traits, which have differing levels or degrees, people are normally classified into one of two traits; introvert or extrovert. Most individuals have been exposed to the concept of personality traits in the form of someone having a Type A or Type B personality. Type A personality is typically impatient and potentially hostile, while Type Bs describe someone who is calm and understated. Personality types are normally determined by taking a fairly lengthy multiple choice test.

Trait theory was initially introduced by Gordon Allport and is the belief that the individual qualities of a person are what determine his or her behavior. Allport also believed these traits can be measured on dimensions or scales, with each one representing a characteristic or trait of that person. Traits are considered to be relatively stable components of our personalities. While there are a potentially infinite number of traits that could define an individual, most research puts the number between three [69][213] and five [134][135]. The three factor model contains the traits of extraversion, neuroticism, and psychoticism while the five factor model contains openness, extraversion, neuroticism, agreeableness, and conscientiousness. Both approaches extensively use self-report questionnaires to gather the

information needed to determine the relevant trait information. Factor analysis is the statistical method used to interpret the scores of the tests by computing the minimum number of characteristics needed to discover the truest assessment of that personality.

Although a great deal of psychological research was compiled in the area of behavioral modeling, we have seen little work mapping psychological behaviors, traits, and types into the realm of computers except in one area. A computer related field that has received significant contributions from the area of psychology is in human-computer interaction (HCI).

2.4.1 Human Computer Interaction

HCI is the study of interaction between people (users) and computers. Because of its distinct yet intertwined human-computer relationship, it is often regarded as the intersection of computer science and behavioral sciences. On the machine side, areas such as computer graphics, operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, cognitive psychology, and human performance are most often used. Early focus in HCI involved user modeling for the purposes of personalizing a user's experience with a computer system. This work ranged from defining user stereotypes [73][178][63] which, combined with user inputs and actions, could aid the system to predict what the user would want to do or see next, to the definition of generic user models which could be used as templates to build specific user profiles of an individual [109][179]. While work in these areas never really met the expectations of the field, numerous contributions were made regarding the identification of computer-based user characteristics and traits which can be extracted for the purposes of this research.

Chapter 3

Cyber Behavior Observables

In this chapter, we outline the types of cyber observables and the pre-processing steps required by each before the Knowledge Discovery phase of our methodology can be performed. We differentiate observables as being *online* (browser based) or *offline* (stored on an individuals computer). Figure 3.1 provides an overview of the basic types of online and offline observables, required pre-processing steps, and the use of these observables in this dissertation. In the remainder of this chapter, we will address each of these observables in detail

Online Data	Pre-Processing	Used in Thesis
URLs	Sessionization	Section 6.3
Search terms	Sessionization, stemming, spell check	Sections 6.2, 6.3, 6.4
Page content	Keyword extraction, stop word removal, stemming	Section 6.3

Offline Data	Pre-Processing	Used in Thesis
Bookmarks	URL extraction	No data available
Cookies	URL extraction	No data available
History Files	URL extraction, sessionization	Section 6.3
Text-based Files	Keyword extraction, stop word removal, stemming	No data available

Figure 3.1: Overview of the online and offline observables, pre-processing steps performed on each, and their use in this dissertation.

and describe the pre-processing required for each. As depicted in Figure 3.2, the input and pre-processing of these observables represents the first step of our behavioral modeling methodology.

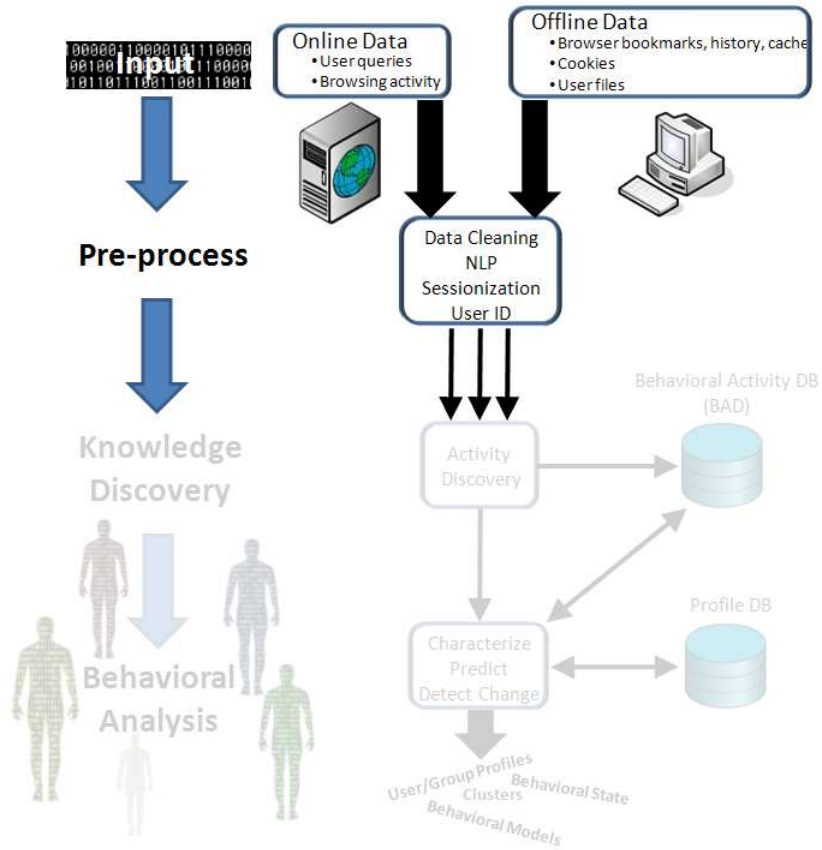


Figure 3.2: Graphical representation of the input phase of our behavioral modeling methodology.

3.1 Online Data

Our widespread reliance on the Internet for everything from research, to hobbies, to shopping, has made *online data* a valuable resource for gathering information about a person. As

reviewed in Section 2.1, a great deal of research is compiled in the profiling and analysis of users based on these activities. One of the shortcomings seen across the spectrum of this research is the fairly narrow focus regarding the actual traffic analyzed (i.e. only obtaining data from specific domains or capture sources).

Our research defines *online data* as information generated from any user initiated web browser based activity. We further classify online data as being either *static* or *interactive* in nature. An example of static data are web pages where the user has little to no interaction with the site. The content of these web pages cannot be changed or manipulated and simply contain information for the user to view or read. *Interactive* data is seen in web sites containing a great deal of unformatted, fluid content making them difficult to interpret using automated techniques. Examples of *interactive* sites include forums, webmail, shopping, banking, and a number of community based sites. Because of their dynamic nature, sites containing interactive data are labeled from the overall activity being performed and not the input entered by the user. For example, a user checking their online Yahoo mail account at `http://us.mg1.mail.yahoo.com/dc/launch?.rand=7arngjeoqlceq` will be identified as using a mail tool by iteratively parsing the domain and extracting `http://mail.yahoo.com`. The actual page content (i.e. e-mail messages, forum posts, finance information, etc.) will not be interpreted or used.

The mechanism used to collect *online data* is irrelevant to this research. As long as each request is associated with a unique user (i.e. no NATed or proxied data), full domain information is maintained, and timestamp information is kept, client based collection, server logs, or passive sniffing (at appropriate locations in the network) all represent valid means to capture this information. The raw *online data* itself is categorized in two broad groups; HTTP GET requests and page content.

GET Requests

As defined in Section 2.1, an HTTP GET request is generated any time a user requests a new web page. This information contains the URL associated with the website being sought and a timestamp documenting when it occurred.

In addition to “traditional” GET requests for specific web pages, search engine queries are also embedded in this data. Whether it is to find more information about a topic for personal interest or for research leading to some future action (i.e. move, new job, new purchase, etc.), search queries provide an abundance of relevant data for behavioral modeling. Search engines currently hold five of the top ten positions as the most popular web sites on the Internet [9] and OPA [157] provides statistics showing web searching as being responsible for 5% of a user’s online time. A web search query is a user-generated request for information to a web search engine to satisfy a specific informational need. While search queries are generally unstructured and often ambiguous in content, they represent a specific focus of the user at a given period of time. Manning et al [127] define three categories of web search queries.

- Informational - covers a broad topic (e.g., colleges or data mining) for which there might be thousands of relevant results.
- Navigational - seeks a single website or web page of a single entity (e.g., Dartmouth or facebook).
- Transactional - reflects the intent of the user to perform a particular action, like purchasing a car or downloading a screen saver.

While navigational queries are of interest from the standpoint of seeing where the user goes, the data itself is really little more than an online bookmark of sorts an individual can use

to find a given site of interest. Informational and transactional queries normally reveal the user's actual interests and intent. Our research does not explicitly classify searches into these categories, however these definitions provide an excellent breakdown of the type of information obtained through different queries and are important to understand when analyzing search traffic.

Whether it is to find more information about a topic for personal interest or for research leading to some future action (i.e. move, new job, new purchase, etc.), search queries can provide a great deal of data relevant to behavioral profiling.

Page Content

While web content is the textual, visual or audio substance associated with a particular web page, this research is only concerned with textual content. When required, page content will be retrieved by downloading (in an automated or manual fashion) a local copy of the web page's information. Data retrieved in this manner is in HyperText Markup Language (HTML) which is then parsed (see Section 3.3) to extract textual content.

3.2 Offline Data

Whereas *online data* examines a user's activity while on the Internet, *offline data* aids in depicting a user via information stored or having the appearance of being stored (i.e. network file systems and shares) locally on their computer.

The amount of personalized data kept on an individual's computer system is staggering. Web browsers, client-based applications, and operating systems store volumes of information about our preferences, past actions and activities. The Firefox web browser (3.6.3 as of this writing) [144], the second most used web browser on the Internet [147], tracks so

much information that a small local database is used to store and track the wide range of user activities and preferences. In addition to standard bookmark data (i.e. bookmark name and associated URL), data is kept and dynamically updated on the number of times a bookmark was visited, the most visited bookmarks, recent bookmark additions, and timestamp information from when the bookmark was added. Web browsers also leave remnants of an individual's browsing patterns and interests in the form of cookies, browser history, and cache files. These files track where a user went on the web and when they were last there. Many operating systems also store various information on recently accessed files and applications.

The last of the offline data sources of consequence are text-based files. Text-based files come in a variety of formats depending on the operating system and application being used, but all share the common theme of being predominately textual in nature. Word processing documents, Portable Document Formats (PDFs), presentations, spreadsheets, and Postscript files stored on a user's machine contain a wealth of information about a given individual. Even rudimentary contextual analysis of these files will reveal much about subject's interests, hobbies, demographics, and employment.

Temporal information is accessible with all of the offline data types mentioned thus far. Bookmarks generally record date/time information when they are created, and many browsers keep a history of the last access. Textual-based files have timestamps relating to the last time the file was Modified, Accessed, or Created (commonly referred to as MAC times). By installing a monitoring agent on a user's machine to track temporal data for previously labeled files, it is possible to create an activity profile including everything from the action a user is performing (web browsing, word processing, reading PDF, etc.), to the context of the action (web browsing computer security sites, creating a Word document on

intrusion detection systems, etc.).

While the inclusion of *offline data* offers a more comprehensive picture of a user's actual computer usage, this type of analysis also requires more intrusive collection techniques and was not performed in this research.

3.3 Pre-Processing

Preprocessing is a critical step in any data mining or knowledge discovery related work. Irrelevant, redundant, unreliable, or noisy data significantly impairs knowledge discovery tools and techniques. The remainder of this section will outline the preprocessing steps taken for input data outlined in Section 4.3.

3.3.1 Queries

The first step in processing queries is the ability to identify them. As stated in Section 4.3, online user data captured is in the form of HTTP GET requests. In general, a web query is just a specially formatted GET request sent in a format a search engine is able to understand and interpret. Because these formats vary from search engine to search engine, initial work in this area only focuses on the top five English based search sites [116]; Google, Yahoo, Microsoft Sites, AOL, and Ask Network . The basic query formats for these five search engines is shown below where `QUERY TEXT` represents the actual search terms.

- Google - `http://www.google.com/search?q=QUERY+TEXT`
- Yahoo - `http://search.yahoo.com/search?p=QUERY+TEXT`

- Microsoft - `http://search.msn.com/results.aspx?q=QUERY+TEXT` or `http://search.live.com/results.aspx?q=QUERY+TEXT`
- Ask - `http://www.ask.com/web?q=QUERY+TEXT`
- AOL -`http://search.aol.com/aol/search?invocationType=comsearch40&query=QUERY+TEXT&do=Search`

Using the above formats, identification of search queries and extraction of query terms is carried out using basic regular expression parsing techniques. Once query terms are identified, the next step is spell checking them. Most search engine sites today provide some form of automatic spell checking for the user. For example, if a user wanted to do a search on Google for *Dartmouth*, but typed in *Dartmoth*, Google will provide a “did you mean” suggestion of *Dartmouth* and will list results related to the correct spelling of the query as well as the incorrect spelling. The user then recognizes their mistake and selects one of the results relating to the correct spelling. While beneficial to the user, this is an incredible hinderance in trying to extract information from raw query data. Since only HTTP GET traffic is being captured, the previous example is represented by a query based HTTP GET request for *Dartmoth* which produces an HTTP GET request for a web page relating to *Dartmouth*. There is nothing in the raw data to represent the “did you mean” correction. This methodology normalizes data via labeling and is unable to reconcile data captured for *Dartmoth*. Left unchecked, the query would be incorrectly labeled or possibly not labeled at all. To address this, all web queries are processed through an open source spell checker (JSpell) with mis-spellings automatically corrected with the “closest” match (usually determined by Levenshtein distance).

3.3.2 Web Page Content

HyperText Markup Language (HTML) is the predominant markup language used to format and present web pages today. By taking advantage of the structure and format of HTML, it is possible to extract the textual information (if any exists) from most web pages. An important distinction should first be made between *content extraction* and *text extraction*. *Text extraction* from a web page is simply the identification of all textual (i.e. non-tag, non-image) information, while *content extraction* is the identification of meaningful content (i.e. no banner text, no advertisements, etc.). *Text extraction* can usually be handled by a simple regular expression based parser capable of interpreting HTML tags. *Content extraction* is not quite as simple.

In addition to the primary content of a site, web pages often contain “noise” in the form of banners, pop-up ads, unnecessary images, navigation links, and menus. Simple text extraction includes all of this irrelevant information, making any type of automated classification or categorization nearly impossible. Research in numerous areas [71][174][137] pertinent to information retrieval and natural language processing attempt to address this.

We currently utilize the *HTML Content Extractor* [207] for content extraction. This tool utilizes a combination of a Document Object Model (DOM) based approach and regular expression parsing to identify and extract relevant content. The Document Object Model [223] is the industry standard for creating and manipulating in-memory representations of HTML or XML content. Research done by [81][128] and others demonstrate how this model is used to identify key content within a web page. Once content is extracted, stemming and stopword removal are accomplished.

Stemming is a common practice used in information retrieval and natural language pro-

cessing to reduce the various inflection and derivations of a word to a common base form. In practice, a stemmer will take words such as *stemming* and *related* and transform them to their base form of *stem* and *relate*. Stemmers are used by most major search engines in order to increase the accuracy and efficiency of their search results. As an example, a user searching for *racecars* would also want results related to *racecar*. To address this, search engines automatically check for a stemmed version of the query against a stemmed and unstemmed version of the data store to maximize results. We currently stem all query terms using a Java implementation of the Porter stemmer [169].

Stop words are lists of extremely common words of little value in helping select documents matching a user query. These inconsequential words are therefore excluded from the vocabulary entirely. While general stop lists exist, most are custom created per domain by sorting the terms by frequency (the total number of times each term appears in the document collection), and selecting the most frequent terms [127]. We use a standard natural language processing stoplist [192] customized by adding common web specific terms (i.e. web, page, copyright, img, etc.).

3.3.3 Clickstreams

As described in Section 2.1, a clickstream is a timestamped historical log of URLs visited over a given period of time. In the context of web mining, these URLs are normally all from the same domain or web site. This research does not limit the scope of the definition and includes all (to include queries) relevant HTTP GET requests generated by a specific user. Relevant GET requests are those generated relating to the intended site the user wishes to go. When a user types a URL in a browser window or clicks on a hyperlink to a site, a HTTP

GET request is generated by the browser to pull back the content of that site. Depending on the content of the page, a large number of additional GET requests are generated to retrieve images, banners, or advertisements associated with the page. Based on the capture method, these “garbage requests” may or may not be present in the data. A client-based capture mechanism could be built into a web browser to collect only “primary” GET requests made by a user. If not pre-filtered, these garbage requests introduce excessive noise and can distort the data. A number of methods exist to minimize the impact of this type of data, but few do a thorough job of removing it. To eliminate this problem, only browser history files are used as a clickstream source. These files contain just the URLs visited and none of the noise associated with *online* based collection.

3.3.4 Sessionization

A user session represents one of the most basic levels of behavioral abstraction from *online data*. We define a *session* as a time ordered sequence of page views by a single user occurring during a period of *activity* by the user. A user is considered *active* during periods of time they are generating HTTP GET requests with a web browser. We identify *activity* through the act of *sessionization*. *Sessionization* deals with segmenting all of a user’s browsing transactions into individual periods of *activity* by the user. These activity sessions are determined in a static or dynamic manner. A statically defined activity session is one in which periods between GET requests do not exceed a predefined static time period. For example, if the time period is set at thirty minutes, a session would consist of all consecutive page requests by said user with no more than a thirty minute separation between any two contiguous requests. This “time out” based heuristic is a common pre-processing step in Web Usage

Mining applications [120]. Clickstream pre-processing, as defined previously, must occur before static sessionization takes place. Static sessionization is straightforward and easily implemented given accurate timestamp information is available.

A dynamic activity session “looks” at the same content the user did. A per user or general case reading speed is defined or calculated from the data and used to estimate the approximate amount of time a given user spent reading a particular page. If the time limit between page views plus some delta is exceeded, a session break is defined. The dynamic nature of web site content makes this a difficult task since the page data retrieved for analysis is often significantly different than the page data viewed by the user. This, combined with the large amount of variability in determining a user’s word per minute reading speed or if a user is even actively reading the content, make dynamic activity sessions a very difficult measure to define and implement.

Due to their wide spread use and simplistic employment, static sessions were selected for use by default in this research.

3.3.5 Text-Based Files

Compared to the complexity of web page content extraction, pre-processing offline text is a simplistic procedure. We currently utilize various aspects of Apache’s Tika [14] toolkit to extract content and metadata from Microsoft files (Excel, Word, Powerpoint, Visio, and Outlook), compressed files (gzip, bzip2), and other formats such as Extensible Markup Language (XML), Hypertext Markup Language (HTML), java class files, java jar files, plain text, PDF, Rich Text Format (RTF), tar archive, and ZIP archive. Once complete, the same stemming and stop word removal procedures used for web content are applied to the text,

further minimizing noise and maximizing content.

3.3.6 Keyword Extraction

A large portion of this work relies on the ability to normalize disparate data through labeling. The current labeling process (described in Chapter 4) requires keywords be identified and extracted from all *offline* and *online* text based data. Keyword extraction is a task to identify a small set of words, key phrases, or key segments of a document best summarizing the meaning of the text. While a large corpus of research exists on various extraction methods [129][7][187][90], we have developed our own keyword extraction tool using a combination of phrase extraction and term weighting techniques outlined in [8].

3.3.6.1 Delicious Tags

In addition to machine generated keyword extraction just described, we also make use of Folksonomy data from Delicious [54]. Delicious is a social bookmarking website where users can store and organize website URLs of interest. By assigning concise tags describing the sites, users can easily search both their own and others bookkmarks for relevant or related sites. Research [7] suggests that most folksonomy words have a higher semantic value than keywords extracted using generic or proprietary keyword extraction techniques. As of November 2008, Delicious reported over 5.3 million users and 180 million unique URLs saved [88]. Initial results shown in Section 4.4 indicate this type of data to be an excellent source of input for our classifier and provides very good classification results.

3.3.7 Browser-Based Data

Although most browsers (and even differing versions of the same browser) store browser generated data in various formats, all tend to collect and track the same fundamental information; URLs and timestamps.

Bookmarks and browser history represent the most basic types of browser-based data. Bookmarks are used in almost all modern web browsers to store URLs of interest for a user to reference at a later time. Browser history is used by a user to “trace their steps” to sites they may have found of interest. While various tools and techniques are needed to extract information from these data sources, once URLs and timestamps are identified, no additional pre-processing is required.

Unlike bookmark and history files, cookies do require pre-processing in order to extract relevant data. As stated in Section 2.1.1.1, a cookie is a small text file used to track user’s browsing activity on individual web sites. Cookies are located on the user’s machine and contain information pertaining to the domain the cookie is associated with. They may be stored in database files or individual text files depending on the browser used. Employing basic regular expression matching, it is straightforward to extract the domain information. This data is important as it may contain a large number of URLs visited which were not listed in bookmarks, history, or cache.

3.4 Summary

In this chapter we have described the types of *online* and *offline* cyber observables which may be used within our behavioral modeling methodology and the steps needed to pre-process this data so it is in a usable format. In Chapter 4 we take the outputs of this phase

of our methodology (keywords, URLs, and search terms) and classify them based on the most representative activity being performed.

Chapter 4

Behavioral Activities

While the terms *online* and *offline* data are concise labels for our cyber-based observables, the output from Chapter 3 consists of a number of disparate behavioral descriptors in the form of keywords, URLs, and queries. In this chapter, we outline the various aspects of the Knowledge Discovery (KD) phase of our behavioral modeling methodology (see Figure 4.1). The goal of this phase is to normalize the inputs via labeling to allow a comprehensive and holistic analysis of all cyber-based behavioral attributes. These labels serve as activity descriptors and represent a key component to our behavioral modeling approach. Due to the diversity of the input types, a combination of supervised learning and data mining techniques are employed to instantiate this activity labeling process.

The first portion of this chapter describes our activity ontology on which we will map these labels. The second half of the chapter describes the algorithms used to transform cyber inputs into activity labels and concludes with an empirical evaluation of its accuracy.

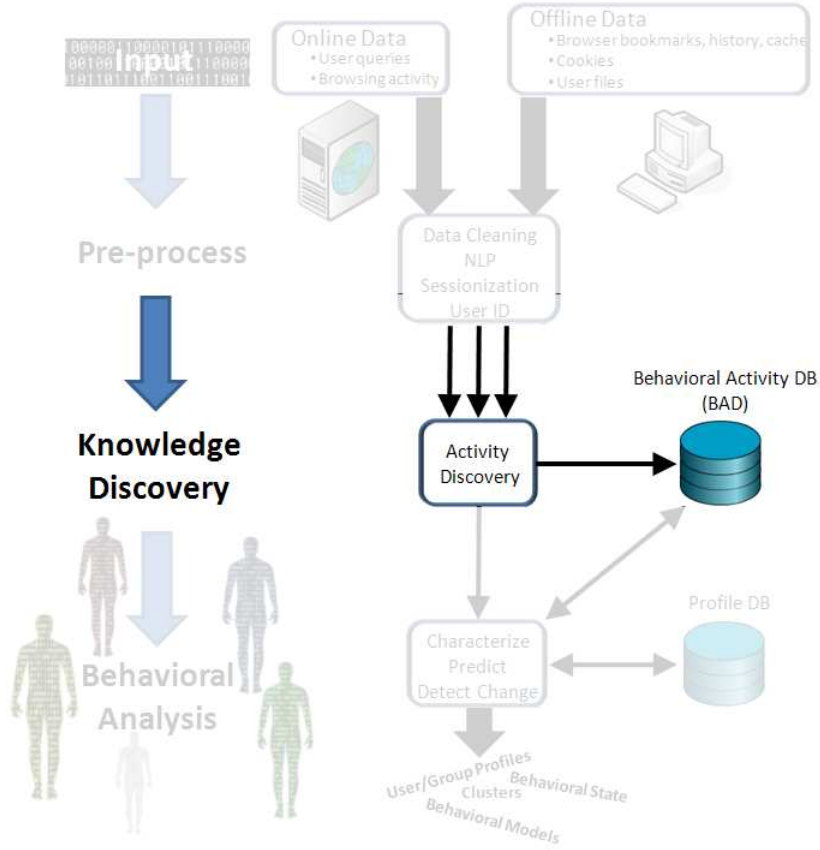


Figure 4.1: Knowledge discovery phase of our behavioral modeling methodology.

4.1 Activity Ontology

While automated approaches exist in topic modeling and clustering to extract labels from data sets based on the information content, most techniques used do not rely on any sort of fixed labeling structure [138][215]. That is to say, labels are generated from key words and phrases making up the text and are not constrained to a pre-determined set of labels. This type of approach can be useful from a human perspective of being able to view and comprehend these descriptive terms, but from an engineering standpoint complicates the normalization of our data and may in fact produce n distinct labels given n distinct objects. Therefore, the choice was made to use a static set of labels to which input data could be

mapped to. The challenge with this approach is finding a representative set of category labels with the depth and flexibility required.

4.1.1 Open Directory Project

The Open Directory Project (ODP) [57] serves as the basis of our activity ontology. The Open Directory Project, also known as Directory Mozilla (DMOZ), is arguably the largest, most comprehensive human-edited directory on the World Wide Web. A number of popular search engines and portals including AOL Search, Netscape Search, Google, Lycos, and HotBotIt, use ODP data to power their directory services. It is constructed and maintained by a global community of volunteer editors. As of June 2010, there were 4,529,962 web sites classified by 85,505 editors in over 590,000 categories. Each URL is categorized based on guidelines defined by the site [153]. URLs with similar content are grouped at a high level category, while lower level sub-categories further define and specify each site. For example, category information for `http://www.dartmouth.edu` would return the following categorical description: *Reference/Education/Colleges and Universities/North America/United States/New Hampshire/Dartmouth College*.

The directory itself is organized as a tree, where each node has a title defined by its location within the directory (e.g., *Reference/Education*). There are seventeen top-level nodes within DMOZ; *Adult, Arts, Business, Computers, Games, Health, Home, Kids and Teens, News, Recreation, Reference, Regional, Science, Shopping, Society, Sports*, and *World*. Each of these top-level nodes has one or more subcategory layers dependent on the diversity of information in each category. This hierarchically labeled structure serves as the basis for our normalization process. Each URL within the directory contains a title and short de-

scription (average number of words per description is 14.89) of the web site. Again using dartmouth.edu, the descriptive information found in DMOZ for this site is as follows:

The smallest of the Ivy League colleges. Located in Hanover, NH.

In addition to the URL, category, and description information, DMOZ data also contains an underlying relatedness link structure referred to as “See also” data. For certain categories, DMOZ editors will identify one or more categories which are contextually related to the category in question. For example, under *Shopping/Health* there are “See also” entries for *Business/Healthcare/Products and Services* and *Health/Resources/Consumer Information*. This data is useful for determining relevance or relatedness of a site to a specific topic. Both the data and the “See also” link structure are available to download and use from the DMOZ web site.

The hierarchical nature of ODP has some significant advantages and disadvantages related to its labeling flexibility. If high level analysis of data is required, it is possible to simply use the the top-level nodes as labels and ignore the remaining category information. We can then modify the “scale” as needed to introduce increased levels of specificity and further delineate the data. For example, if interested in only the top three category labels, Dartmouth would be labeled as *Reference/Education/Colleges and Universities* instead of the full title listed previously. This is beneficial in areas such as clustering and classification, where the level of granularity can significantly impact effectiveness and efficiency.

Disadvantageous aspects of the hierarchical structure include increased noise introduced by some of the category definitions. The *World* category is a composite of the whole directory in other languages. While of potential use for future research, we currently ignore this category completely. The *Kids and Teens* category is similar to *World* in that it reproduces

the directory but with a focus on the young. The *Regional* category contains approximately one third of the data and is dedicated to listing English language sites about various geographical regions of the world. This category starts with continents and countries and distinguishes all the way down to towns and landmarks. Similar to *Kids and Teens*, many of the sub-categories of *Regional* are the same as those defined in upper echelon ODP categories. Under *Regional/North America/United States/New Hampshire*, one finds categories *Business and Economy*, *Health*, *Recreation and Sports*, *Science and Environment*, and *Shopping*, all of which are similar to the top level categories *Business*, *Health*, *Sports*, *Science*, and *Shopping*. While these distinctions are useful for determining geographic preferences of an individual, for general labeling this level of granularity is unnecessary. For this reason, the sub-categories for *Regional* and *Kids and Teens* were manually moved into the appropriate higher level categories. In addition, a new top level category, *Government*, was created from the *Regional* data. This category consists of approximately forty thousand entries dedicated to national, state, and local government agencies. Lastly, the *News* category is an elaborate tree comprised almost entirely of news stories. The topics of these stories span the spectrum of all of the categories in DMOZ, introducing an excess amount of noise from a clustering and classification standpoint. To counter this, the category is reserved for labeling URLs only.

4.1.2 Blacklists

While we have just described the robust coverage of DMOZ and how it serves as the foundation for our activity ontology, it is lacking in one particular area; inappropriate content. Other than the *Adult* category, which focuses on URLs containing explicit sexual content,

the rest of the sites listed are fairly benign in subject matter and are appropriate for all ages. Though the focus of our behavioral modeling is not deviancy, there are a number of significant behaviors in this realm which we would like to capture (violence, hacking, crime, etc.). We make use of content filters to capture this type of data.

Content filtering is commonly used by organizations, such as offices and schools, to prevent computer users from viewing inappropriate web content, surfing on unwanted web sites (non school or work related), being exposed to network attacks, or wasting bandwidth (streaming media sites). Filtering rules are typically implemented via software on individual computers or at a central point on the network such as a proxy server or gateway router. One method to control access to unwanted categories of websites is through a URL blacklist database. Like DMOZ, these databases are organized by category and contain lists of prohibited URLs and IP addresses. Unlike DMOZ, these databases do not contain the title, description, or “See also” information. Administrators can use the categories to block *types* of information by simply instantiating the appropriate category. These database categories and associated URLs are used to strengthen our ontology and provide the information needed to describe behaviors falling outside the social “norm”. Utilizing several open source URL blacklists [216][111], we map the URLs and categories onto our activity ontology, adding approximately 1.2 million unique entries and twelve new lower level category labels.

The finalized list of top level categories used is as follows; *Adult, Arts, Business, Computers, Games, Government, Health, Home, Kids and Teens, News, Recreation, Reference, Regional, Science, Shopping, Society, and Sports*. High level descriptions of these top level categories can be found in Appendix B.

4.1.3 User-defined Ontology

While our current research revolves around using the DMOZ structure and associated data as our normalization source, this is by no means the only approach to this problem. Because user requirements may demand categories constrained to a specific domain, the remainder of this section describes additional repositories of data as well as techniques to organize internal data for this approach.

4.1.3.1 Human Generated Directories

Human generated hierarchical data sets are the preferred training source as humans are still the best at interpreting and categorizing textual data. Most human edited directories are updated and maintained by a cadre of editors abiding by stringent guidelines dictating how data is organized. In addition to ODP, other website categorizations include AboutUs.org [1], Best of the Web Directory [33], BUBL [39], and Yahoo! Directory [232] to name a few. The Medical Subject Headings (MeSH) taxonomy [151], US Patent Classification [161], and the Library of Congress Classification [51] are additional sources of data usable in whole or part for training purposes.

One disadvantage to using a human edited corpus is that if you don't have one or can't find one for your domain, you must create one. Large corpora of human categorized data are difficult to come by due to the monetary and time costs incurred with the definition and maintenance of these projects. If external sources of data are located and domain appropriate, the next hurdle is accessing the raw data. Of the previously listed sources, only ODP allows users to download a full data capture for external applications. The last disadvantage in using human edited data involves human error and biases. While humans

are the best at analyzing and interpreting textual data, their susceptibility to mistakes and biases may lead to the misclassification of information.

Disadvantages aside, human edited data is still the best training source for this work and should be considered if at all possible.

4.1.3.2 Machine Generated Directories

Due to the overhead and difficulties associated with finding or creating and maintaining one's own human edited directory, the next choice is a computer generated one. The identification and definition of groups of similar data is a subject explored extensively under various disciplines the last three decades. A significant disadvantage seen to machine categorization is its inability to interpret terms having multiple meanings based on context, including polysemy and homonyms. For example, the word *bank* can reference an institution used to hold money or the side of a river. Any machine generated categorization task should involve some sort of manual verification and monitoring. In the absence of user-generated categorizations, machine-made datasets are often the only alternative. Document classification tasks can be divided into two types; supervised and unsupervised.

In supervised text categorization, the objective is to learn classifiers from examples or training sets. The three most widely studied and effective algorithms for supervised text categorization are k nearest neighbor (k -NN), Naive Bayes, and support vector machines (SVM). While a great deal of research exists using these algorithms [208][95][195][234][45][108][149], all three rely on the existence of pre-categorized training data. While some research has been performed using Naive Bayes [150] and SVM [218] on unlabeled training data, results are not yet comparable to their labeled counterpart. Another limitation to most of the research done in this area is that results are usually “flat” in nature and not meant to deal

with hierarchical results. While not a requirement of this research, we believe a hierarchical dataset offers the greatest amount of flexibility and extensibility.

In unsupervised learning, the machine receives inputs only with no supervised target outputs of any kind provided. One of the more simple, but well explored forms of unsupervised learning relative to text categorization is clustering. Clustering is the process of finding natural groups in unlabeled data. Like supervised learning, a number of well studied algorithms exist [12] in the realm of text-based categorization, but most pertinent to this work is research in the area of hierarchical text clustering.

Much of the research in hierarchical clustering [72][76][56][188] follows the concept of incremental hierarchical conceptual clustering. The premise is, given a sequence of instances and their associated descriptions, find: clusterings that group instances into categories, an intentional definition for each category that summarizes its instances, and a hierarchical organization for those categories. In terms of this research, its value lies in its ability to both hierarchically cluster training data and incrementally build descriptive behavioral models. The Java machine learning toolkit Weka [224] includes the hierarchical clustering algorithm COBWEB [72].

Topic Models are yet another unsupervised learning technique and are covered in detail in Section 5.1 and 5.1.1. By treating documents as a mixture of topics (given a topic is a probability distribution over words), topic modeling provides a simple mechanism to analyze and label copious volumes of text. Contextual clues within the text are used to connect words with similar meanings and distinguish words with multiple meanings. Topic models were used successfully in [132][182][79] as a means to effectively and efficiently extract key concepts from text.

4.1.3.3 Folksonomy/Collaborative Based Directories

While folksonomy and collaborative-based data sources could fall under the Human Edited Section of this paper, we list them separately as they are normally not defined specifically for the purpose of categorizing data.

A folksonomy, also referred to as collaborative tagging, social classification, social indexing, and social tagging, is a user-generated taxonomy developed from the collaborative creation and management of tags to annotate and categorize content. A tag is simply meta data in the form of one or more keywords or terms best describing the content in question. Delicious [54], one of the most well known folksonomies on the web today, allows users to bookmark and share web sites of interest. Unlike DMOZ, these sites use a non-hierarchical “tag-based” mechanism to index and store data. Users “tag” bookmarks using keywords most representative of the site. These tags can then be searched directly (to find other related sites) or can be displayed for an individual URL to gain a quick synopsis of the web page. Research in this area [7][226][6][156] has demonstrated the significant benefit to using folksonomy-based data for classification purposes. This is verified in our initial results (see Section 4.4) where tags from the Delicious folksonomy are used for automated web page classification.

Collaboration is a recursive process involving two or more people or organizations working together to achieve intersecting goals through shared knowledge, learning and building consensus[139][38]. Collaboration does not, in general, require leadership and can sometimes bring better results through decentralization and egalitarianism. Collaborative software (sometimes referred to as groupware) is software designed to help people involved in a common task to achieve their goals. Wikis are one of the more common online collaboration tools

with Wikipedia being one of the largest and most well known collaborative projects in existence. Wikipedia is written by teams of volunteers from all around the world, hosting more than 75,000 active contributors working on over 13,000,000 articles in 260+ languages. Anyone with Internet access can amend Wikipedia articles. Since its creation in 2001, Wikipedia has grown into one of the largest reference web sites in the world.

Although Wikipedia has what can best be described as a “shallow” hierarchy, its nodes contain very high-quality, noise free articles offering an excellent training source for document categorization and classification tasks. Collaborative data sets such as Wikipedia offer tremendous potential in the areas of text categorization and knowledge management. Much recent research [83][164][75][196][74] has focused on finding ways to leverage such information. While we have not yet experimented using collaborative data sets like these, we see no reason why such a data set would not work with our current classification methodology. In lieu of human categorized data, folksonomy and collaborative based sources appear to be an equal if not superior data source choice.

4.2 Activity Assignment

With an activity ontology defined, the next step in the process is the actual assignment of activities to input. Referring back to our behavioral model of Section 1.2, the goal of activity assignment is to instantiate the *activity trees* of the user. These trees describe the observable actions of the user and are fundamental for the extraction of user behaviors.

Information retrieval and supervised learning techniques have been used extensively for the purposes of labeling text-based (queries and keywords) information via classification [203][97][202] and categorization [108][149]. Results of this work vary but tend to be either

constrained to a small set of labels, have narrow domains (keyword or training set focused in a specific area), or require large amounts of additional information obtained from external sources.

A web or link directory is a information repository on the World Wide Web specializing in linking to other web sites and providing categorization of those links. The categorization is usually done manually and determined by the content of the whole web site rather than one page or a set of keywords. Category labels are predefined and strict guidance is in place to determine how they are used. Yahoo!Directory [232] and the Open Directory Project are two of the most well known web directories on the Internet. Although web directories are not considered search engines, they are searchable. The primary difference is when querying a web directory, instead of retrieving a list of related web sites, it returns a list of web sites by category and subcategory. Like web search engines, results are normally found by examining an index and providing an ordered list of the most significant entries based on a pre-defined similarity measure. While this provides a simple means to assign labels to input, a significant drawback is the lack of a relevancy score without which there is no way to determine if the first result returned is at all representative of the input entered.

Search result clustering engines like Carrot2 [158] and clustering search engines such as Clusty [219] automatically group (according to various similarity measures) and label (based on the predominant theme of the group) search results. Clusters are presented to the user sequentially via the number of results per cluster. Like web directories, no relevancy measure is given per result or per category. As a result, a cluster at the top of the list may in fact contain the most irrelevant entries.

To address the clustering and relevancy issues encountered in web directories and cluster engines, we use a combination of the two approaches to create a classifier that both clusters

and ranks results dependent upon on the significance of the cluster.

4.2.1 Training Data

While certain online data, such as URLs, may be labeled via a direct lookup in our ontology to extract the appropriate category information, a mechanism must exist to label URLs not in the ontology and that data which cannot be searched for directly. This becomes a text classification problem where given our input (keywords or search terms), we must assign the most relevant category based on some supervised (deducing a function from training data) or unsupervised (deduction based on no a priori knowledge) learning method. Because our activity ontology provides such a large training resource, we utilize supervised learning techniques to accomplish this task.

The DMOZ URL, title, description, and category information serve as the primary training data for mapping inputs to category labels. Since our blacklist data does not contain title or description information, it is not used for training purposes. In order to minimize noise, only unique URLs (those not listed in multiple categories) and their associated title and description are used. With these constraints in hand, our training set contains approximately 4.4 million categorized items from seventeen top level categories totaling approximately sixty million words.

4.2.2 Storage

The training data is stored in an inverted index data structure to more efficiently and effectively access the text and category information. As outlined in Section 2.2.1, this type of data structure is ideal for storing a mapping from content to location and is well suited

for full text search. While a number of commercial and open source indexing tools exist, each with various strengths and weaknesses, we use Lucene [53] as our main storage and retrieval mechanism. Lucene is an information retrieval library originally written in Java, whose primary function is full text indexing and search.

In Lucene, objects are stored as *Documents*, where a *Document* is a collection of *Fields*. Each *field* corresponds to a piece of data either queried against or retrieved from the index during a search. If one were storing e-mail messages in a Lucene index, they could define fields for To, From, Subject, and Body for the meta data associated with each portion of the message. For this research, the document name is the DMOZ category label and the fields contains the DMOZ title, description, and URL data. All field information but URL are first stemmed and had stop-word removal performed prior to storage in the index as described in Section 3.3. Figure 4.2 is a graphical representation of both a generic (left hand side of figure) and ontology instantiated (right hand side of figure) representation of the Lucene index structure. Figure 4.3 is high-level diagram outlining the training and storage aspects of the classifier.

Once the training data is stored in the index, queries against the index return relevancy scored results much like a standard search engine. The objects being scored are *Documents* and scoring works on *Fields* (as defined above). Lucene scoring uses the Vector Space Model (VSM) of Information Retrieval to assess how relevant a given Document is to a given query. VSM scores each document's relevancy by determining how many times a query term appears in a document relative to the number of times the term appears in all the documents in the collection. Lucene offers additional capabilities and refinements in support of boolean and fuzzy searching, but essentially remains a VSM based system.

Lucene allows score "boosting" in three different areas; Document level boosting, Docu-

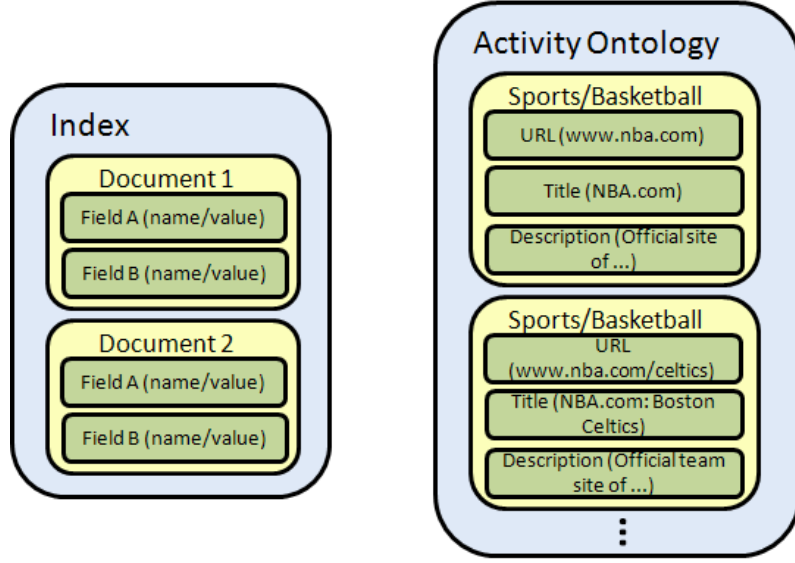


Figure 4.2: Generic (left hand side) and ontology instantiated (right hand side) representation of the Lucene index structure.

ment Field level boosting, and query level boosting. By default, all *Documents* and *Fields* in a Lucene index are weighted the same with a boost factor of one. By changing a *Document* or *Field* boost factor, it is possible to have Lucene weight these attributes with more or less emphasis with respect to other *Documents* or *Fields* in the index.

The score of a query q for document d correlates to the cosine-distance or dot-product between document and query vectors. A document whose vector is closer to the query vector in that model is scored higher. The score is computed as follows:

$$score(q, d) = \cos(\theta) = \frac{\vec{V}_q \cdot \vec{V}_d}{|\vec{V}_q| \times |\vec{V}_d|} = \frac{1}{\sqrt{\sum_{t \in q} idf(t)^2}} \times \sum_{t \in q} (tf(t, d) \times idf(t)^2 \times \frac{1}{\sqrt{\text{num of terms in field } f}})$$

where

- $tf(t, d)$ is the term frequency factor for the term (t) in the document (d),
- $idf(t)$ is the inverse document frequency of the term,

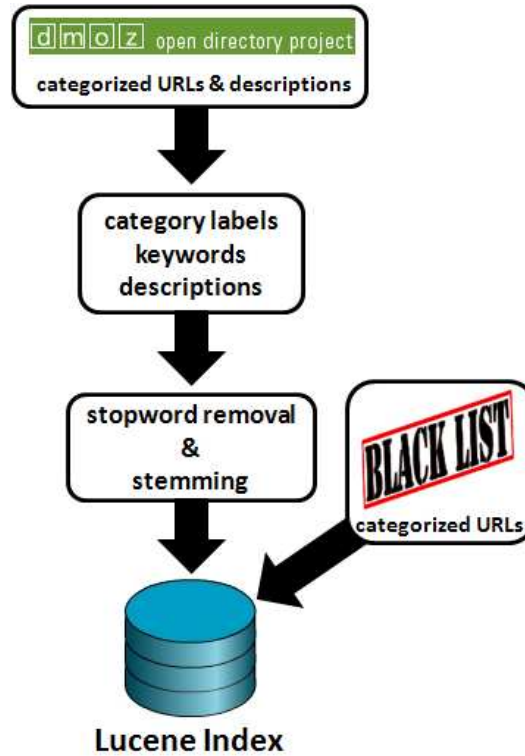


Figure 4.3: Graphical depiction of our index instantiation using DMOZ and blacklist data.

4.2.3 Labeling Algorithm

With an index and scoring mechanism in place, we can now query our ontology for the most appropriate labels to our cyber-based inputs (keywords, URLs, and search terms). Using the query *dartmouth college*, below are the top twenty category labels returned. Note, for readability purposes, some categorical information was cropped on certain results.

SCORE	CATEGORY

5.4868	Computers/Computer_Science/Academic_Departments/North_America/United_States/New_Hampshire
5.4868	Society/History/Academic_Departments/North_America/United_States/New_Hampshire
5.4075	Reference/Education/Colleges_and_Universities/North_America/United_States/New_Hampshire
5.4075	Society/Gay,_Lesbian,_and_Bisexual/Student/Colleges_and_Universities/North_America/United_States
5.4075	Recreation/Climbing/Organizations/North_America/United_States
4.8497	Reference/Education/Colleges_and_Universities/North_America
4.8010	Reference/Maps/Libraries
4.8010	Arts/Literature/Authors/M/Milton,_John/Works/Of_Education
4.8010	Society/Ethnicity/The_Americas/Indigenous/Native_Americans/Education/Academic_Departments
4.8010	Science/Social_Sciences/Sociology/Academic_Departments/United_States/D

```

4.1151 Reference/Education/Colleges_and_Universities/North_America/United_States/New_Hampshire
4.1151 Arts/Literature/Authors/M/Milton,_John/Works/Areopagitica
4.1151 Reference/Education/Colleges_and_Universities/North_America/United_States/New_Hampshire/Dartmouth
4.1151 Society/Philosophy/Academic_Departments/North_America/United_States
4.1151 Reference/Education/Colleges_and_Universities/North_America/United_States/New_Hampshire/Dartmouth
4.1151 Science/Math/Academic_Departments/North_America/United_States
3.8798 Sports/Cricket/ICC/Associate_Members/United_States/College_and_University
3.8798 Reference/Education/Colleges_and_Universities/North_America/Business
3.7728 Reference/Education/Colleges_and_Universities/North_America/United_States/New_Hampshire/Dartmouth
3.7728 Reference/Education/Colleges_and_Universities/North_America/United_States/New_Hampshire/Dartmouth

```

In addition to category labels, results contain a relevance score of the query to the document. A naïve approach to labeling would assign the category label of the first result returned. For the example just shown, *dartmouth college* would be labeled as *Computers/Computer Science/Academic Departments/North America/United States/New Hampshire*. While not an incorrect classification, the logic of the approach is flawed in a number of ways. First, the second result is scored exactly the same as the first. Lucene places an arbitrary ordering on results containing the same relevance score. A second issue is the frequency of categories. The *Computers* result is only listed once in the top twenty while the top level category *Reference/Education* is listed eight separate times. While *Computers* has the single highest score, the cumulative results of *Reference/Education* far outweighs this. An algorithm is needed which utilizes both the ordering of the results, as well as the frequency of a given label when determining the most appropriate category label. A k -nearest neighbor (k -NN) like algorithm with a dynamic k is used to meet this requirement

The goal of traditional k -NN is to use a labeled training set to classify unknown target data into one of a fixed number of classes. For each element of the target data set, the k (where k is determined ahead of time) most similar training items are identified by calculating how “close” each member of the training set is to the target element being analyzed. Closeness is normally calculated using a measure such as Euclidian distance. The k -closest data points are then used to determine the most common (majority rules) class label which is

then assigned to the target element. Should two or more class labels occur an equal number of times, the k -NN test is run again with k equal to $k-1$ (one less neighbor). This process continues until there are either no ties or until k is equal to one.

Figure 4.4 is a graphical depiction of how the algorithm works. The green circle represents

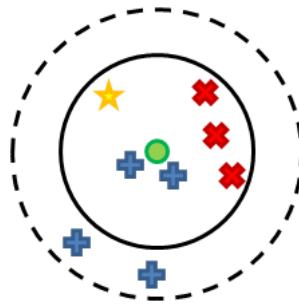


Figure 4.4: k -NN algorithm for k equal to six (solid circle) and k equal to eight (dashed circle).

the target data. The items within the solid circle represent the training data when k is set to six. From the figure, it is apparent the common training element belongs to the X class. Therefore, the target data is labeled as a red X. The items within the dashed outer circle represent the training set when k is set to eight. The most common class label is now the blue + sign, signifying the target data is best represented as a blue plus.

Applying this algorithm to our *dartmouth college* example, if we set k equal to twenty and only examine the first three levels of category labels, then the chosen class (or in this case, label) would be *Reference/Education/Colleges and Universities*. While the algorithm addresses both ordering and frequency, there are limitations to using k -NN with our ontology. The first concern is the individual scores are not explicitly taken into account. To demonstrate this, below are the top four results for the query *java coffee*.

SCORE	CATEGORY
4.0189	Shopping/Food/Beverages/Coffee_and_Tea/Coffee/Espresso
3.8427	Shopping/Food/Beverages/Coffee_and_Tea/Coffee/Espresso
2.3447	Computers/Programming/Languages/Java/Applets/Collections
2.3447	Computers/Programming/Languages/Java/News_and_Media/Books

Setting k equal to four and utilizing the k -NN algorithm, we are presented with a tie even though *Shopping/Food* dominates *Computers/Programming* from a cumulative score perspective. To address this, we modify the algorithm to select the class with the highest cumulative score vice highest number of occurrences. The change still takes into account the number of instances of a category, but more importantly, it factors in the relevance of the individual results.

A second, and more difficult problem with using k -NN is determining k . When using k -NN, the best choice of k depends greatly on the training data. In general, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. While various heuristics exist to approximate k [62], initial testing on our ontology yielded poor results. Bayesian inferencing is used to estimate an optimal k based on an iterative evaluation of the results returned.

Bayesian inference uses a numerical estimate of the degree of belief in a hypothesis prior to any observations and calculates a numerical estimate of the degree of belief in the hypothesis after observations have occurred. This process can then be repeated as additional evidence is obtained. Bayesian inference is based on Bayes' theorem, which adjusts probabilities given new evidence in the following way:

$$P(H|E) = \frac{P(E|H) P(H)}{P(E)}$$

where

- H represents a specific hypothesis, which may or may not be some null hypothesis
- $P(H)$ is the prior probability of H , inferred before new evidence, E , became available
- $P(E|H)$ is the conditional probability of seeing the evidence E if hypothesis H happens to be true
- $P(E)$ is the marginal probability of E and represents the apriori probability of witnessing new evidence E under all possible hypotheses
- $P(H|E)$ is the posterior probability of H given E

Bayes' theorem allows us to determine how much new evidence is required to alter a belief in a hypothesis. We use this concept to calculate how many results must be obtained before we are confident we have enough data to accurately classify a given input.

We represent each category label as a binomial distribution. By iterating through each result returned one at a time, we count the number of m_i "successes" when category i is present and the number of n_i "failures" for results not containing category i . Our belief about the category of interest (i) is stated as the prior distribution, $p(i)$. For some special choices of the prior distribution $p(i)$, the posterior takes a convenient form. This occurs when the posterior distributions are in the same family as the prior probability distribution. When this happens, the prior and posterior are called conjugate distributions, and the prior is called a conjugate prior for the likelihood [31]. The Beta distribution is the conjugate prior to the binomial distribution [31]. Because of this, given $p(i)$ is a beta distribution with parameters m_0 and n_0 , then the posterior is also a beta distribution with parameters $m + m_0$ and $n + n_0$. Although our proportion of interest is multinomial in nature, we can relax the problem and simplify the mathematics by modeling our observations as binomial (as just described). We use this to calculate confidence interval for our posterior distribution of each category i .

By starting with a uniform Beta prior (shape parameters $\alpha = 1, \beta = 1$) and then updating the shape parameters for each new observation, we calculate an upper and lower confidence bound on the posterior distribution for each category evaluated. Upper and lower bounds are calculated by taking the inverse cumulative probability. For example to calculate the upper and lower bounds for a 95% confidence interval, one solves the below two equations for x and y :

$$P(X < x) = .975$$

$$P(X < y) = .025$$

When the upper bound minus the the lower bound ($x - y$) for each category is within a pre-determined *K_threshold* value, k is set and the category with the highest cumulative score is chosen. *K_threshold* values may be determined based on known accuracy constraints or may be calculated empirically given labeled training data is available (see Section 4.4 for an empirical evaluation example).

Using just the top two category levels from the *Dartmouth College* results as an example, each distinct category is represented as a binomial distribution where successes and failures are counted based on the category appearing or not appearing in subsequent results. When the upper bound minus the lower bound for each category (in this case *Computers*, *Society*, *Reference*, *Recreation*, *Arts*, *Science*, and *Sports*) is within our threshold, k is set and no further results are examined. The category with the highest cumulative score is chosen as the label for the given input. Figure 4.5 shows the results from our *Dartmouth College* example, where n represents the total number of samples, y the number of successes, μ' the sample mean, UB the upper confidence bound, LB the lower confidence bound, *Interval* the difference between the upper and lower confidence bound, and *Score*

the cumulative score for this category. For this example, we use a confidence interval of

Category	n	y	μ'	UB	LB	Interval	Score
Reference/Education	20	8	0.409091	0.496735	0.31994	0.176795	34.0279
Arts/Literature	20	2	0.136364	0.192911	0.073963	0.1189483	8.9161
Computers/Computer_Science	20	1	0.090909	0.135972	0.039431	0.0965416	5.4868
Society/History	20	1	0.090909	0.135972	0.039431	0.0965416	5.4868
Society/Gay,_Lesbian,_and_Bisexual	20	1	0.090909	0.135972	0.039431	0.0965416	5.4075
Recreation/Climbing	20	1	0.090909	0.135972	0.039431	0.0965416	5.4075
Reference/Maps	20	1	0.090909	0.135972	0.039431	0.0965416	4.801
Society/Ethnicity	20	1	0.090909	0.135972	0.039431	0.0965416	4.801
Science/Social_Science	20	1	0.090909	0.135972	0.039431	0.0965416	4.801
Society/Philosophy	20	1	0.090909	0.135972	0.039431	0.0965416	4.1151
Science/Math	20	1	0.090909	0.135972	0.039431	0.0965416	4.1151
Sports/Cricket	20	1	0.090909	0.135972	0.039431	0.0965416	3.8798

Figure 4.5: Selection of the most relevant category from our *Dartmouth College* example with a confidence interval of 80% and a $K_threshold$ value set to 0.2 (we want to be 80% confident we are within 20% of the true mean for each category).

80% and a $K_threshold$ value of 0.2 (we want to be 80% confident we are within 20% of the true mean for each category). After evaluating the first twenty results, we see that our $K_threshold$ is satisfied by all categories and determine, based on the cumulative scores, *Reference/Education* to be the most representative label. While ranked number one in our initial results, *Computers/Computer_Science* is tied for third using this approach.

Below is a basic outline of our modified k -NN approach.

1. Query index to obtain list of relevant labels
2. Use Bayesian inferencing to determine value of k
3. Determine the number of distinct result labels
 - (a) Calculate the cumulative score of each label
4. Choose label with the highest score

Figure 4.6 is a graphical representation of our modified k -NN algorithm.

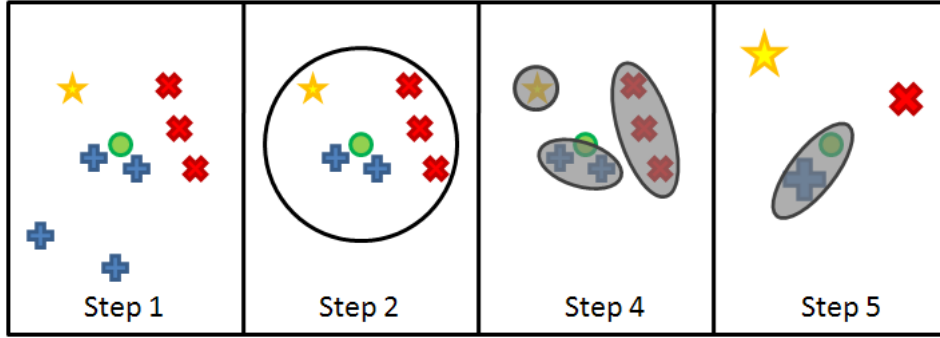


Figure 4.6: Modified k -NN algorithm based on Bayesian inferencing

Like the traditional k -NN algorithm, the accuracy of our approach can be severely degraded by the presence of noisy or irrelevant features [26][145]. This further justified and strengthened our desire to have a training set based on short descriptions and keywords.

4.3 Activity Inputs

As outlined in Chapter 3, after pre-processing has occurred inputs needing to be labeled will be in the form of keywords, URLs, or queries. Once data is appropriately labeled, it is stored in the Behavioral Activity Database as depicted in Figure 4.1 for further analysis. Currently this is implemented via a MySQL database.

4.3.1 Keywords

Keywords are the most simple to label as they are already in the proper format for our classifier. Because not all keywords have equal importance, we weigh keywords based on their normalized term frequency. Just as Lucene allows for indexed fields to be weighted differently, it also provides a mechanism for input terms to be weighted as well. Using the caret (^) symbol, one can assign a different weight to each input term. To give the term

Dartmouth twice the weight as the term *College* in the query *Dartmouth College*, one simply formats the input as follows:

$$Dartmouth^2 College$$

Once properly weighted, keywords are input directly to our classifier and labels generated with no additional processing required.

4.3.2 URLs

The process for labeling URLs can best be described via the data flow diagram of figure 4.7. The first step is to determine if the URL in question is in our activity ontology. Because

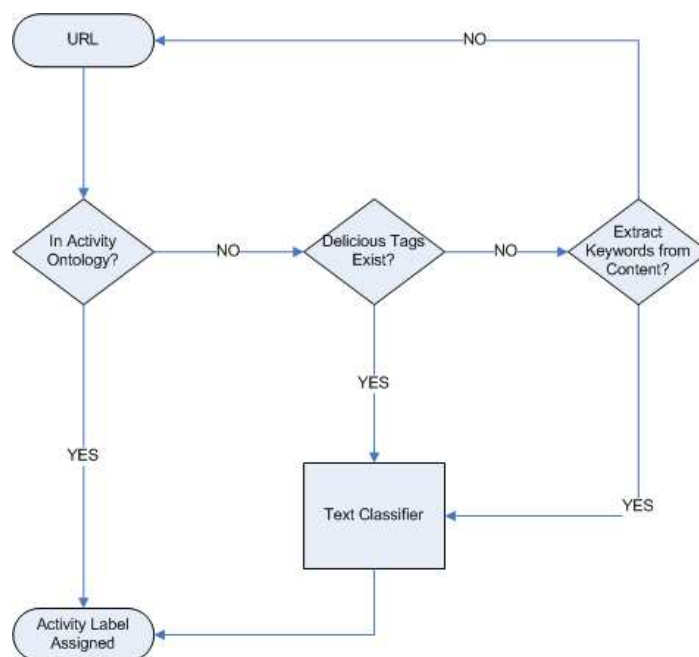


Figure 4.7: Data flow diagram depicting how URLs are given activity labels.

only unique URLs (single URL per category) are contained in our ontology, if the URL is present, it is automatically labeled. If the URL does not exist in the ontology, Delicious is

queried to determine if user-generated tags exist for the URL. As we will show in Section 4.4, Delicious tags offer a significant increase in classification accuracy vice using machine generated keywords. If Delicious tags exist, they are input into the classifier and an activity label is assigned. If no tags exist, keywords are automatically extracted from the content of the page using the techniques described in Section 3.3.6. If insufficient text is available to obtain a statistically representative set of terms, the URL is deemed unclassifiable and the next URL is processed. If keywords can be extracted, they are input to the classifier for labeling.

4.3.3 Search Engine Queries

Search engine queries are processed using a combination of the techniques used for keywords and URLs. Unlike traditional query classification where one has to determine the most appropriate category for a query based on search terms alone, our input contains the URL visited based on the query results. Because of query ambiguity, we use the URL visited as the primary classification item and the query terms for contextual clarification. Using the keyword weighting scheme described in Section 4.3.1 we weigh URLs (or URL keywords dependent on the results of Section 4.3.2) twice that of the query terms to give preference to the site visited following the query. We currently have no way to test the effectiveness of this approach as no data sets currently exist with human categorization based on search queries and the associated URL followed.

4.4 Activity Assignment Accuracy

We first test the accuracy of our labeling algorithm using manually generated keywords from web page text. The initial test set consisted of three hundred random URLs selected from each of the seventeen top level categories of our ontology. All textual content from each URL was downloaded and keywords were identified and weighted according to Section 4.3. For our initial testing, at least ten keywords had to be extracted from the text in order for it to be considered for classification. Further testing is needed to determine the optimal number of keywords for this task. Of the 5,100 possible results, 3,706 URLs were used. The 1,394 entries not used were either no longer valid URLs (web site did not exist) or did not have enough textual information to extract ten statistically significant keywords. The 3,706 test URLs, titles, and descriptions were then removed from the classifier index to ensure results were not skewed. Each URL was classified into its highest level category (i.e. *Sports* vice *Sports/Basketball*) using the keywords extracted.

To determine the `K_threshold` value, the data was run through the classifier ten times setting a static confidence level of 0.9 and varying the upper/lower bound interval between 0.1 and 0.55. The interval which yielded the best results for the data was 0.4, a 73.1% accuracy for the top level categories. Figure 4.8 graphically depicts the results of this test. A similar experiment (same categories minus *Adult* and 500 random samples per category vice 400) performed in [172] achieved 65% accuracy using the Rainbow [131] implementation of k -NN with a static $k=30$ and 73.1% accuracy using SVM^{light} [96] implementation of support vector machines. Upon closer examination of classification results, we noticed a number of “near miss” misclassifications. For example, a web site manually labeled *Arts/Crafts/Glass/Blowing* was classified with our algorithm as *Shopping/Crafts/Glass/Hot*

Glass/Blown. To take these similarities into account, we made use of the DMOZ “See also” information as described in Section 4.2.1. By examining incorrect results and determining if they were “See also” results for the target category, we were able to identify 40 additional results (increasing our accuracy to 74.2%). By relaxing this constraint slightly and not just looking at exact matches, but those incorrect classifications which were “one off” a related category, 101 additional results from our basic classification were obtained, thus boosting our accuracy to 75.8%. By “one off” we are referring to a scenario when a “see also” category was *Home/Consumer Information/Computers and Internet/Internet/Access Providers/Free* and our classification was *Home/Consumer Information/Computers and Internet/Internet/Access Providers*. Based on our “one off” principle, these categories would be a match.

A second test was run to evaluate how well our algorithm would perform using Delicious tags (see Section 3.3.6.1). The DeliciousT140 dataset [237] was created in June 2008 with data retrieved from Delicious and the Web. The data set consists of 144,574 unique URLs, all of them with their corresponding social tags retrieved from Delicious. 6,471 URLs were identified in this data set containing ten or more Delicious tags which were also present in our activity ontology. As with the first test, data was evaluated by the classifier ten times varying the confidence interval between 0.1 and 0.55 to determine an optimal *K_threshold*. The value which provided the highest accuracy for the data was 0.3, yielding an 80.9% accuracy for the top level categories. A comparison of the accuracy results is shown in Figure 4.8. While the *K_threshold* values are not exact, their relative closeness allow us to define a general threshold which can be used for either manually generated keywords or human generated tags. In keeping with our initial test, evaluation of the “See also” criteria was also performed. Correct results increased by 63 and 220 accounting for accuracy percentages of 81.9% and

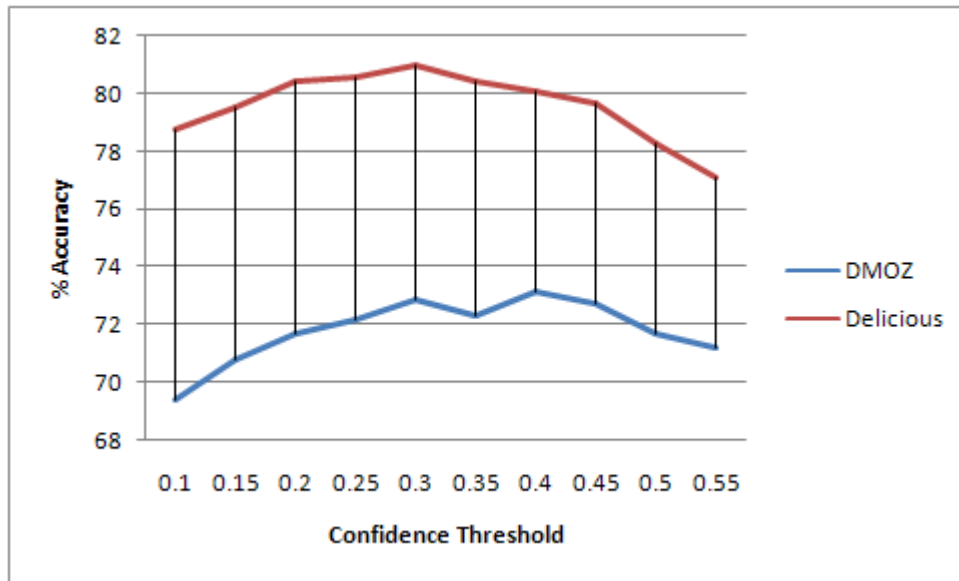


Figure 4.8: Accuracy of using manually generated keywords (DMOZ) versus human generated Delicious tags.

84.3% respectively.

Further examination of the results from both tests revealed an interesting trend in the data. With a $K_threshold$ set at .4 for our first test, when k was calculated to be 5, our prediction accuracy was 92%. This k factor was present in 1,429 of the 2,710 (53%) of the correct results. Using this same threshold in the Delicious test, when k was calculated to be 8, the prediction accuracy was 97.3% and accounted for 1,778 of the 5,183 (34%) correct results. The benefit to being able to identify when our classifier will have very accurate results allows us to learn from previously un-categorized data. New data classified with one of these k values can be with high confidence categorized correctly and added to our training index allowing our classifier to learn and grow as it encounters more data.

4.5 Summary

In this chapter we have described our activity ontology and the associated algorithms used to map cyber-based observables to hierarchical activity trees. We have demonstrated the accuracy of these algorithms as well as their ability to learn and expand their knowledge base in an automated manner. In Chapter 5, we will detail the analysis techniques used to take the activity trees created in this chapter and extract and analyze behavioral information in a quantitative manner.

Chapter 5

Behavioral Analysis

Behavioral analysis is the most critical phase of our methodology (see Figure 5.1) and is concerned with the application of mathematically derived principles to cyber-based activities for the purposes of characterization, prediction, and change detection in user and group behaviors. In the remainder of this chapter, we define our behavioral model and outline the methods used to analyze and interpret this model in a quantitative manner.

5.1 Behavioral Model

While in Chapter 1 we provided a graphical representation of our behavioral model, we now define this model mathematically. From a purely representational standpoint, we see our problem domain most similar to that of topic models [30][182][132][210]. Topic models are generative models allowing sets of observations to be explained by unobserved groups which describe why some parts of the data are similar. Typically applied to document collections (abstracts, e-mails, etc.), topic models are based on the idea that documents are mixtures of a small number of topics and each word is attributable to one of the document's topics.

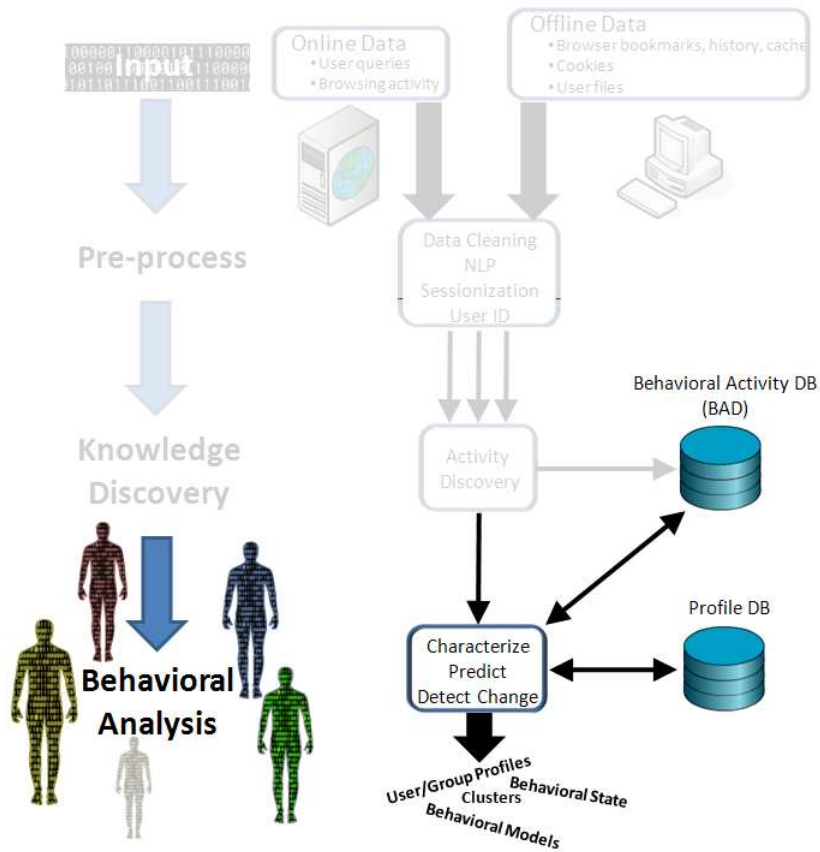


Figure 5.1: Graphical representation of the analysis phase of our behavioral modeling methodology.

In contrast to generative models, statistical inference is a technique used to invert the process just described by inferring the set of topics responsible for generating a collection of documents. We use the same approach in our domain, but map documents, words, and topics to sessions, activities, and behaviors. The table in Figure 5.2 summarizes this correspondence. Using this terminology, behavioral models are based on the idea that sessions

Topic Model Example	Topic Model		Behavioral Model	Behavioral Model Example
Books, pdfs, abstracts, e-mails, etc.	Documents	=>	Sessions	Browsing sessions, query sessions, etc.
Unique words in all documents	Words	=>	Activities	Sports, Shopping, News, etc.
Personal e-mail, work e-mail, etc.	Topics	=>	Behaviors	Buying a car, researching a project, etc.

Figure 5.2: Mapping of the topics model concept of documents being a mixture of topics to our behavioral model of behaviors being a mixture of sessions.

are mixtures of a small number of behaviors and each activity is attributable to one of the session’s behaviors, where a cyber behavior is a probability distribution over activities.

Borrowing from the example presented in [210], Figure 5.3 is a graphical comparison of a generative process (left) versus statistical inference (right) of a topic model in a behavioral context. On the left, the generative process depicts two behaviors (behavior 1 and behavior

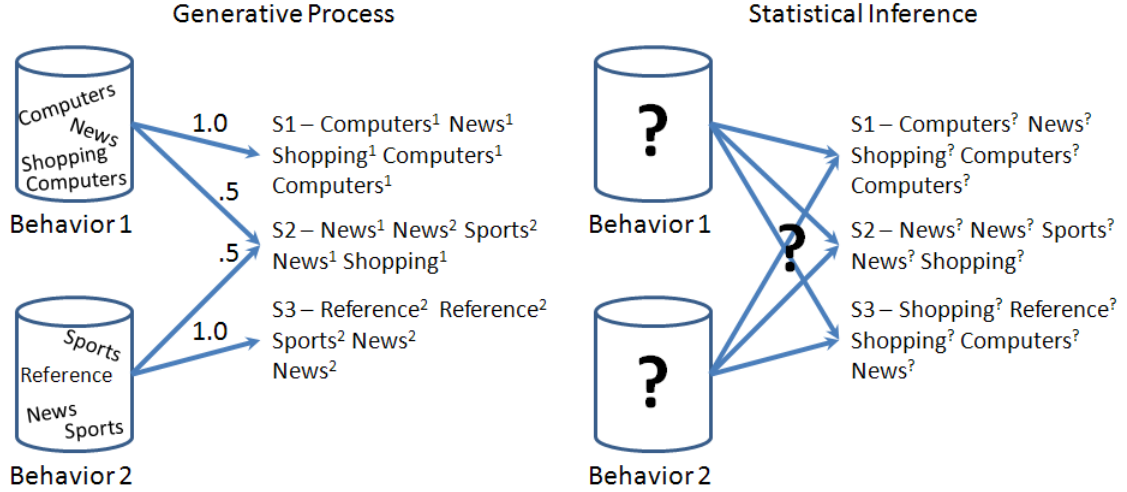


Figure 5.3: Graphical depiction of the generative process (left) and that of statistical inference (right) as it relates to the generation and extraction of cyber behaviors.

2) and the sessions (S1, S2, and S3) generated by each . Both behaviors have a common relationship to *News* but are focused on *Computers* (behavior 1) and *Sports* (behavior 2) related activities. These behaviors are represented as “bags-of-activities” containing different distributions over activities from which sessions can be generated. In this case, session 1 is generated by behavior 1, session 3 by behavior 2, and session 2 is generated by an equal mixture of both behaviors (superscripts in the diagram indicate which behavior was used to sample each activity).

Although implied by our common *News* activity, we emphasize there is no notion of

mutual exclusivity restricting activities to a single behavior. This is an important aspect of the model allowing for a single activity to be represented in multiple behavioral contexts. For example, in behavior one, *News* is related to computer related news and events whereas in behavior two, it is restricted to sporting news.

The right portion of Figure 5.3 illustrates statistical inference. Given the observed activities in a collection of sessions, we wish to know the behavioral model responsible for generating the data. This involves inferring the probability distribution over activities associated with each behavior, the distribution over behaviors for each session, and the behavior responsible for generating each activity.

Figure 5.4 is a generic session/activity matrix used to store our bag-of-activities data. Here S are sessions, A the individual activities, a the activity counts per session, n the

	A_1	A_2	...	A_k
S_1	a_{11}	a_{21}	...	a_{k1}
S_2	a_{12}	a_{22}	...	a_{k2}
:	:	:		:
:	:	:		:
S_n	a_{1n}	a_{2n}	...	a_{kn}

Figure 5.4: Session/activity matrix representation of an individual's cyber activities.

number of sessions, and k the number of activities we are looking to model. Our goal of storing information in this matrix is to then perform a factorization of the sessions/activity matrix into a behaviors/activity matrix and a sessions/behavior matrix (see Figure 5.5). Our bag-of-activities representation tracks activity counts only, not the order of activities

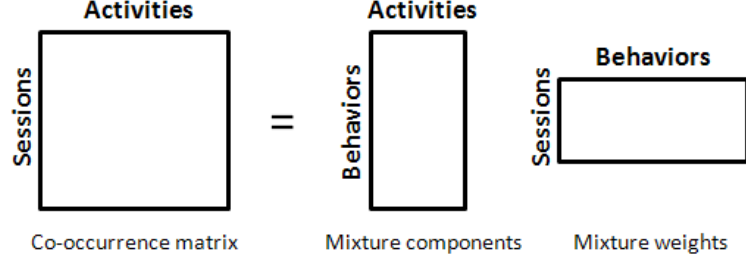


Figure 5.5: The matrix factorization of a session/activity matrix into a behaviors/activity matrix and a sessions/behavior matrix

within a session. We understand activity-order may provide additional behavioral insights. Griffiths et al [80] present an extension of the topic model which takes into account word order. We believe this to be a viable approach to further define behaviors, but leave it as an area for future research.

While various instantiations of probabilistic topic models exist [30][210][27][29], our model remains fundamentally the same no matter which we use; sessions are a mixture of behaviors and behaviors are a probability distribution over activities. Each activity in a session is seen as a sample from a mixture model where mixture components are multinomial $P(a_i|z = Z_j)$ and the mixing proportions are $P(z = Z_j|s_S)$ (where z is a latent behavior and Z_j is one of $1, \dots, Z$ behavioral values). Before going any further, we formalize our definitions of activity, session, user, and behavior as follows:

- Activity – basic unit of discrete data, a , defined to be an item from a vocabulary V where the v^{th} activity in the vocabulary is represented as a^v
- Session – a sequence of N_s activities denoted by $s = a_1, a_2, \dots, a_{N_s}$ where a_n is the n^{th} activity in the session
 - N – the total number of activity tokens ($N = \sum N_d$)
- User – collection of S sessions denoted by $U = s_1, s_2, \dots, s_S$
- Behavior – each behavior z (where Z is the total number of behaviors) is represented

as a multinomial distribution over activities

Borrowing notation from [210], $P(z)$ is the distribution over behaviors z in a session and $P(a|z)$ is the probability distribution over activities a given behavior z . $P(z_i = j)$ is the probability the j^{th} behavior was sampled for the i^{th} activity token and $P(a_i|z_i = j)$ is probability of activity a_i under behavior j . The distribution over activities within a session can now be represented as follows, where Z is the total number of behaviors:

$$P(a_i) = \sum_{j=1}^Z P(a_i|z_i = j)P(z_i = j)$$

To simplify notation, we let $\phi^{(j)} = P(a|z = j)$ refer to the multinomial distribution over activities for behavior j and $\theta^{(s)} = P(z)$ refer to the multinomial distribution over behaviors for session s . Parameter ϕ indicates which activities are “important” for which behavior and parameter θ represents the mixture weights to identify “important” behaviors for a session.

Typically a Dirichlet [31] prior on θ is used in order to estimate the mixture weights. The parameters of the distribution are specified as $\alpha_1 \dots \alpha_Z$ where each hyperparameter α_j is interpreted as a prior observation count for the number of times behavior j is sampled in a session before having observed any actual activities in the session. Normally a symmetric Dirichlet distribution with single α parameter is used to create a smoothed behavioral distribution (the amount of smoothing is determined by α). A symmetric Dirichlet(β) prior is also used on ϕ and represents the prior observation count on the number of times activities are sampled from a behavior before any activity is observed. Depending on the implementation used, various techniques exist to estimate each parameter just described.

In order to capture the dependencies among the parameters, we represent our model in Figure 5.6 using a behavior-based version of the plate notation originally presented in [210]. The arrows in the model indicate conditional dependencies between variables while plates (boxes) represent repetitions of sampling steps (variables in the lower right corner refer to

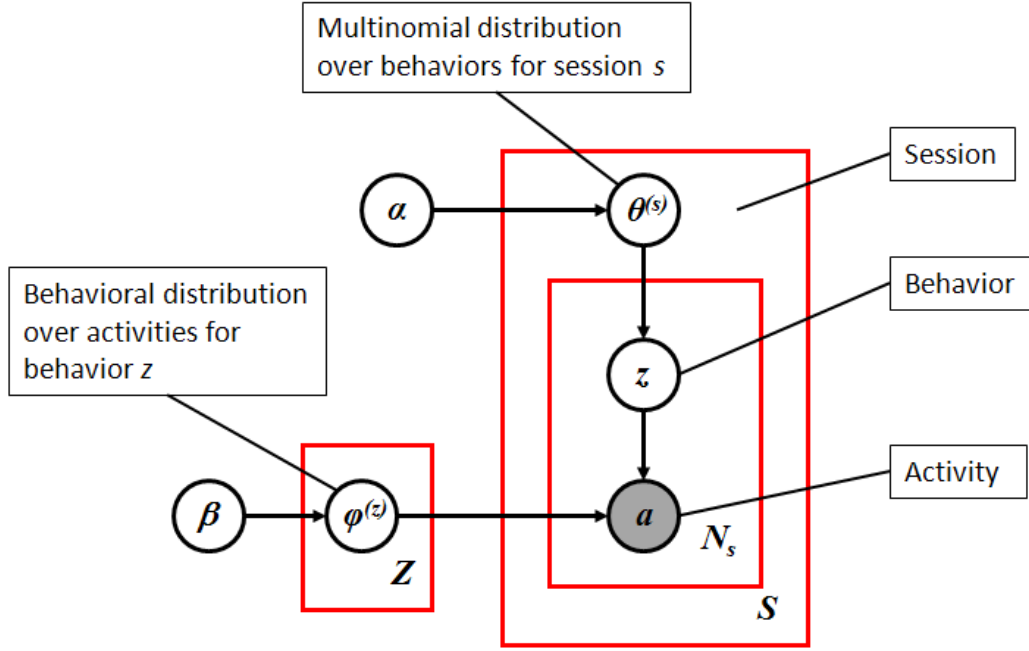


Figure 5.6: Plate notation depicting our cyber-based behavioral model

the number of samples). The activities a (shaded), are the only observable variables in this model while all others (non-shaded), are latent variables. Using this notation, the inner plate over z and a illustrates the repeated sampling of behaviors and activities until N_s activities have been generated for session s . The plate surrounding $\theta^{(s)}$ shows the sampling of a distribution over behaviors for each session s for a total of S sessions, while the plate surrounding $\phi^{(z)}$ illustrates the repeated sampling of activity distributions for each behavior z until Z behaviors have been generated.

5.1.1 Behavioral Extraction

Given a session/activity matrix such as in Figure 5.4, the goal of behavioral extraction is to use the concepts just described to factor the matrix as shown in Figure 5.5 to identify the behaviors represented and the distribution of activities which created them. While a number of algorithm exist to achieve this [30][210], we make use of a sampling based version of Latent

Dirichlet Allocation (LDA) described in [210] and implemented in the Java package Mallet [130].

LDA [30] is a probabilistic generative model used to estimate the multinomial observations by unsupervised learning. Estimating parameters for LDA by directly and exactly maximizing the likelihood of the whole data collection is intractable and therefore requires approximate estimation methods such as variational inference, Expectation propagation, and Gibbs Sampling. We make use of the Gibbs sampling approach in this work.

Gibbs sampling is a specific form of Markov chain Monte Carlo (MCMC) and simulates a high-dimensional distribution by sampling on lower-dimensional subsets of variables where each subset is conditioned on the value of all others. Sampling is done sequentially and proceeds until the sampled values approximate the target distribution. Using this approach, activities are first randomly assigned to behaviors. Repeated iteration over every activity is then performed, choosing a new behavior for the activity based on the current assignments of every other activity within a session and the activities assigned to each behavior. The probability of assigning an activity a in session s to behavior z is given by

$$p(z|s, a) \propto \frac{\alpha_z + N_s^t}{\sum_z' (\alpha_z' + N_s^{z'})} \frac{\beta_a + N_z^a}{\sum_a' (\beta_a' + N_z^{a'})}$$

where N_s^z is the number of times behavior z appears in session s and N_z^a is the number of times activities of type a have been assigned to behavior z . Using LDA, the hyperparameters α_z and β_a are generally constants, reflecting symmetric, uninformative priors.

During the initial stage of the sampling process (also known as the burn-in period), the Gibbs samples are discarded because they are poor estimates of the posterior. After the burn-in period, the successive Gibbs samples begin to approximate the target distribution, which in our case, is the posterior distribution over behavior assignments.

Estimates for ϕ' (activity-behavior distribution) and θ' (behavior-session distribution) can now be directly extracted as follows:

$$\phi_i^{(j)} = \frac{N_{z_j}^{a_i} + \beta}{\sum_{k=1}^A N_{z_j}^{a_k} + A\beta} \quad \theta_j^{(l)} = \frac{N_{z_j}^{s_l} + \alpha}{\sum_{k=1}^Z N_{z_k}^{s_l} + Z\alpha}$$

These values correspond to the predictive distributions of sampling a new token of activity i from behavior j , and a new (unobserved) token in session l from behavior j . These values also represent the posterior means of these quantities conditioned on a particular sample z .

The choice of the number of behaviors is an important factor in our approach as too small a choice will result in very broad behaviors and too large a choice will lead to uninterpretable results. A large corpus of research [28][214][182] exists on methods to select the most appropriate number of behaviors, and is beyond the scope of this research. We currently use a combination of Hierarchical Latent Dirichlet Allocation [28] based on nonparametric Bayesian statistics and manual inspection to best determine the appropriate behavior count.

5.2 Behavioral Traits

While the model outlined in Section 5.1 provides a means to describe and extract cyber behaviors, it does not provide a mechanism to represent specific behavioral characteristics of an individual. Adapted from Shannon’s Information Theory [122] and transcribed to a behavioral context in [189], we use n^{th} order models to describe these characteristics and will refer to them hereafter as *behavioral traits*. While not a representation of behaviors in and of themselves, these traits are critical to providing a holistic depiction of both individuals and groups. We will use these models extensively in the remainder of this chapter to identify, filter, and cluster individuals.

A 0^{th} order model is an enumeration of observed cyber activities of an individual or

group. Using our activity ontology (see Section 4.1), an example of a 0^{th} order model would be $\{Sports/Basketball, News, Computers/Programming\}$.

A 1^{st} order model is a listing of the 0^{th} order activities, along with the associated frequency of performing the activity or a posterior distribution describing the probability of engaging in the activity. Using this terminology, as defined in Section 5.1, behaviors are 1^{st} order with respect to activities and sessions are 1^{st} order with respect to behaviors. Figure 5.7 depicts two 1^{st} order representations of the user described in the previous example. The left

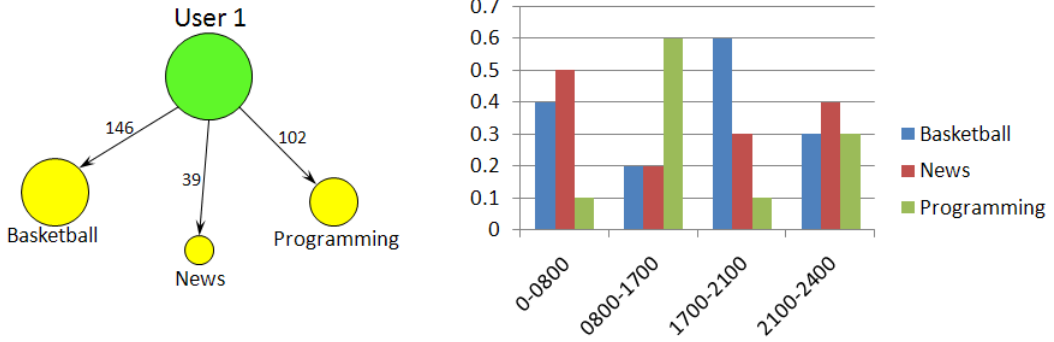


Figure 5.7: Depiction of two 1^{st} order models showing total activity counts (left) and conditional frequencies of activities (right)

hand side of the figure graphically depicts the total counts associated with observing the user performing each activity. The right hand side of the graph is a bar chart showing the conditional frequencies based on time of day of these same activities.

Lastly, a 2^{nd} order model is a probabilistic characterization of activities conditioned on other activities, rather than simply conditioned on environmental properties. Given that behaviors are 1^{st} order with respect to activities and sessions are 1^{st} order with respect to behaviors, if we have a temporal ordering on sessions, then they are 2^{nd} order with respect to behaviors. Building on our example, Figure 5.8 is a representation of a 2^{nd} order model. The figure depicts a Markov model showing the probabilities of performing each activity

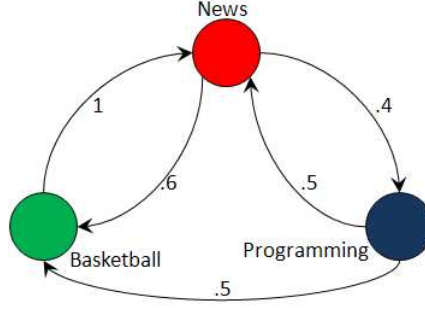


Figure 5.8: Depiction of a 2^{nd} order Markov model showing the probabilities of performing an activity given an activity was just performed.

given an activity was just performed.

5.3 Behavioral Sample Size Estimation

Calculating the amount of data required to represent a variable of interest is critical to determining the reliability and accuracy of one's experiment. Too small a sample lacks sufficient power for detection of a statistically meaningful difference, while too large a sample may add unwanted noise. Frequentist methods specify a null and alternative hypothesis for the parameter of interest, then find the sample size by controlling both size and power. By contrast, the Bayesian approach offers a wide variety of techniques, all of which offer the ability to deal with uncertainty associated with prior information. In the remainder of this section, we outline an approach based on Bayesian bootstrapping to determine the minimum number of observed activities to ensure statistical reliability for the classification and identification of individuals through online actions.

5.3.1 Naïve Estimation

A naïve approach to this problem can best be summarized as exhaustive search. Given a data set D and a classifier C , incremental train/test partitioning of the data is performed

using a holdout procedure (certain amount of data is used for testing and the remainder is used for training) until the classifier yields the highest accuracy. Repeating this process for a large population will eventually produce the average sample size needed to adequately classify a user. While this approach is not recommended, it provides a baseline from which we can compare results for our formal sample size determination method.

5.3.2 Empirical Estimation

A significant problem with the naïve approach is the use of a global cutoff value. For example, *User A* visits web site *X* one hundred and ninety nine times and web site *Y* once, whereas *User B* visits web site *X* one hundred times and site *Y* one hundred times during the same period. We clearly require less data to accurately train a classifier for *User A* than we do for *User B*, but using a naïve approach does not take advantage of this fact. To address this limitation, we expand our basic definition of a user as defined in Section 1.2 from a collection of S sessions denoted by $U = s_1, s_2, \dots, s_S$ to a more detailed 2^{nd} order representation. We describe a user by a probability distribution over k possible activities with probabilities p_1, \dots, p_k . Applying this to the standard multinomial probability distribution over words formula from IR, we have the following model of a user:

$$\frac{L_u!}{af_{a_1,u}!af_{a_2,u}!\dots af_{a_k,u}!} P(a_1)^{af_{a_1,u}} P(a_2)^{af_{a_2,u}} \dots P(a_k)^{af_{a_k,u}}$$

Here, $af_{a_1,u}$ is user u 's activity frequency (af) for activity a_1 , $L_u = \sum_{i=1}^k af_{a_i,u}$ is the length of the session, and k is the size of the activity vocabulary. Frequency in this model is calculated as the number of unique activities per session over all sessions. We are interested in unique activities per session vice total activities per session as we are more concerned in what a user is interested in over time rather than a potentially very focused, yet fleeting interest during a short period. For example, a user may browse to the same four stockmarket websites every day for a month. On a given day this individual learns of a new flu virus

warning. Consequently, the user visits hundreds of health related sites to research this information. Examining monthly activity totals, the user would have 120 counts associated with the finance activity (4 counts per day for 30 days). In one day, however, the user accumulates 400 counts associated with the activity category health. While of obvious interest on that particular day, health issues are not representative of the user over time, but examining cumulative totals, these numbers dominate finance counts. Using unique activities per session, the user would accrue 30 counts associated with finance (1 unique activity per day) and 1 count associated with health (assuming all activities happened in one session).

By defining random variable x_i as the number of times activity i is uniquely observed per session over n sessions, and k equal to the total number of activities observed, we represent a user by the activity vector $U = (x_1, \dots, x_k)$. Our interest is in determining how much data is required before probabilities p_1, \dots, p_k associated with the user's k activities converge to within some tolerance of their true value.

One of the first resampling methods for estimating the confidence interval of a sample with minimal assumptions is the bootstrap [65]. Bootstrapping is the practice of estimating properties of an estimator (such as its variance) by measuring those properties when sampling from an approximating distribution. One standard choice for an approximating distribution is the empirical distribution of the observed data. In the case where a set of observations can be assumed to be from an independent and identically distributed population, this can be implemented by constructing a number of resamples of the observed dataset (and of equal size to the observed dataset), each of which is obtained by random sampling with replacement from the original dataset. The bootstrap was modified and put in a Bayesian context as the Bayesian Bootstrap [183] where the likelihood of the data is assumed to be multinomial with unknown probabilities w_i and the prior is the noninformative $\prod_{i=1}^n w_i^{-1}$. Rubin [183] shows that the posterior distribution of the w_i 's given the data is the

n -dimensional Dirichlet distribution with all parameters equal to 1 (Dirichlet(1, ..., 1)). The Dirichlet is the conjugate prior distribution for the parameters of the multinomial, such that the posterior Dirichlet's parameters equal the sum of the count data, x , and the parameters from the prior Dirichlet, α as represented below.

$$p(\theta|x + \alpha) = p(x|\theta)p(\alpha)/p(x)$$

Here, x represents the unique activity count data, θ the parameters of the multinomial distribution associated with these counts $(\theta_1, \theta_2, \dots, \theta_k)$, and α the hyper parameters of the prior Dirichlet distribution $(\alpha_1, \alpha_2, \dots, \alpha_k)$. Using the Bayesian Bootstrap, one samples from an n -dimensional Dirichlet distribution.

The key to numerically drawing the w_i 's from the Dirichlet distribution is to use the result from probability theory that a set of n independent gamma random variables divided by its sum has a Dirichlet distribution. In particular, if we let Z_i be a gamma random variable with mean and variance 1, and we draw n independent Z_i 's, then the set of n

$$w_i = \frac{Z_i}{\sum_{i=1}^n Z_i}$$

has an n -dimensional Dirichlet distribution with all parameters equal to 1. We use this to incrementally evaluate credible intervals for the parameters of the multinomial distribution over time. Credible intervals for the parameters are calculated on samples drawn from an appropriate distribution, which in the case of a multinomial distribution is a Dirichlet distribution. Because we are modeling users as a multinomial distribution over activities, we can utilize this approach to calculate the number of sessions required for all activity proportions to be within some threshold of their true value. This forms a stable "snapshot" of the user for a given time period t and most precisely represents the user given the current observations. Calculating the minimal sample size needed to represent an individual is performed in the following manner.

1. Start with a non-informative prior (i.e. $(x_1, x_2, \dots, x_k) = (1, 1, \dots, 1)$)
2. Collect sample (i.e. $(S1_1, S1_2, S1_3, S1_4, S1_5) = (1, 0, 0, 0, 1)$)
3. Simulate draws from posterior (as n independent gamma variables divided by the sum)
4. Calculate credible intervals for each multinomial variable
5. If credible interval for each variable not within “acceptable” threshold, update prior with sample counts and go to step 2
6. If credible interval for each variable within “acceptable” threshold, set prior and create sliding window

The determination of an “acceptable” threshold is application dependent. For clustering on 0^{th} order traits, a 70% confidence of being within 20% of the mean for each parameter may be satisfactory, while individual identification requires much more stringent bounds.

As previously stated, the initial sample size estimation is intended to offer a snapshot of the user, capturing only those behaviors exhibited during a discrete timeframe. An issue which must be addressed in dealing with this approach is overfitting. Overfitting occurs when a statistical model describes random error or noise instead of the underlying relationships. It generally occurs when a model is excessively complex or has too many degrees of freedom. To avoid overfitting in our model, the initial sample size calculated determines the size of a “sliding window”. As new data arrives, old data is discarded in a first in first out fashion. This approach is reliable given the user continues to behave in a manner consistent with the activities performed when the sample size was calculated. As new observations arrive, behavioral changes will cause multinomial parameters to change, exceeding pre-determined confidence thresholds. To account for these changes, we use an adaptive control mechanism to monitor and update our sliding window size. Figure 5.9 is a data flow diagram outlining the steps taken by our adaptive controller. As new observations arrive, the oldest observation is removed. The posterior is then calculated and credible intervals for each parameter

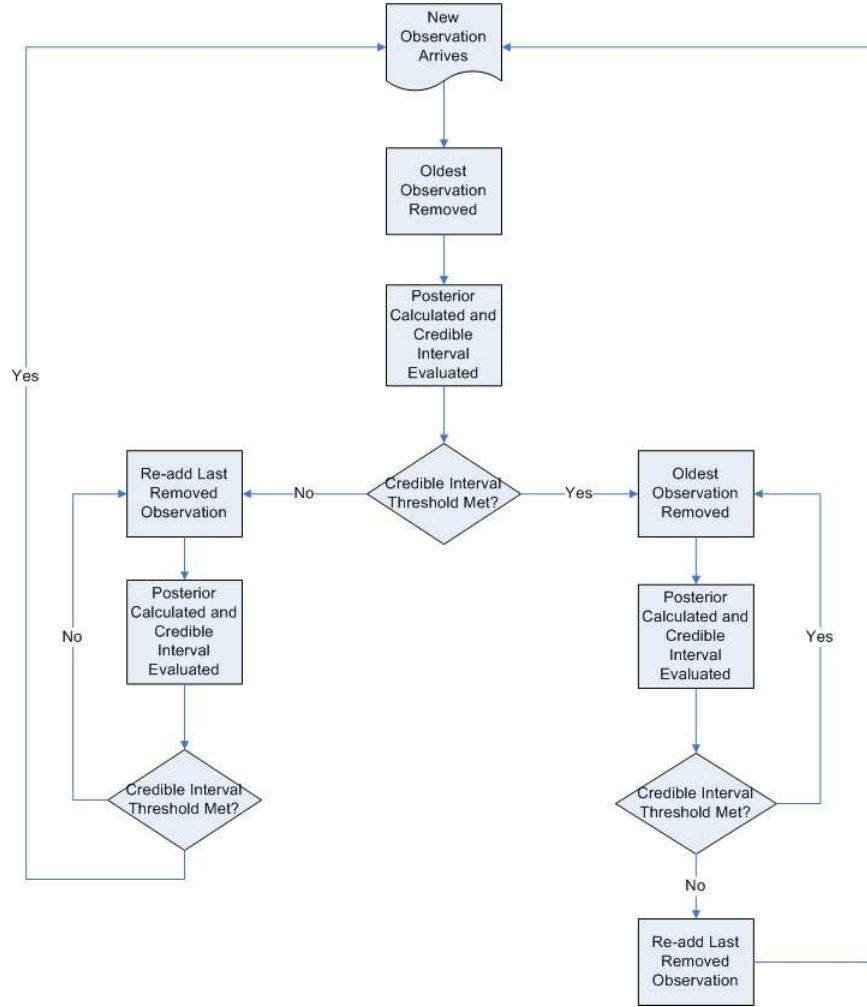


Figure 5.9: Data flow diagram of adaptive control mechanism used to dynamically grow (left branch of decision node) or shrink (right branch of decision node) the sliding window size.

evaluated. If all parameters are within our pre-defined threshold (right branch of decision node), then the controller shrinks the window by removing another observation on the queue. Parameters are re-evaluated and the process continues until the threshold is no longer met. At this point, the last observation removed is re-added and new observations are collected. If all parameters of our initial observation are not within the pre-defined threshold (left branch of decision node), then the controller looks to grow the window until the threshold is attained.

The sample size algorithm outlined in this section is empirically evaluated in Section 5.4.2.1.

5.4 Behavior-Based Characterization

There are two critical components in making characterizations; it must describe unique qualities or peculiarities and be a distinctive trait or mark. Within the scope of this research, the objective of characterization is to identify those behaviors and behavioral traits which may be used to uniquely identify and describe individuals and groups.

5.4.1 Behavioral Profiles

User profiles and the act of profiling can be applied in a variety of different domains and for a variety of purposes. Criminal profiling by behavioral scientists and the police narrows down the suspects in an investigation to those who possess only certain behavioral and personality features. Profiling is also used in the financial sector for fraud prevention and credit scoring and by e-commerce to predict the behavior of different types of customers. We define a cyber-based profile as a collection of behavioral traits used to uniquely identify a specific individual or group of users. A profile refers therefore to the explicit digital representation of a person's identity. We note that a profile represents a *use* of a behavioral model and is not a behavioral model itself.

Cyber profiles can be created for an individual or a group of people. Here individual profiles will describe the characteristics of a single user and group profiles categorize a person based on shared characteristics with a population of interest. Clustering (see Section 5.4.3) is one mechanism to identify group profile characteristics. Contextual and/or temporal information is used to create both individual and group profiles. The following subsections describe these attributes and how they may be used in detail.

5.4.1.1 Contextual Attributes

Contextual attributes can be used to specify or clarify the description of an individual or group based on surroundings, environment, or background information. In the cyber realm, these attributes present themselves in the form of 0^{th} and 1^{st} order models. Occupation, hobbies, or even aspects of one's family life, all provide contextual descriptors which can be used to create a profile. *Sports/Basketball*, *Regional/Colorado*, and *Computers/Programming* may be an adequate 0^{th} order group profile to identify all individuals living in Colorado who are computer programmers and like basketball.

Specificity of the data can range from general, group level characteristics, to very specific individual eccentricities depending on the type and scope of the profile needed. The goal is to find the most representative activities from our ontology, then describe individuals and groups based on the attributes identified. We currently follow a six step process to create a contextual profile.

1. Identify meaningful attributes
2. Search activity ontology categories
3. Search activity ontology descriptions
4. Identify pertinent "See also" categories
5. Instantiate profile
6. Refine

The first step in the process is to identify those attributes most representative of the individual or group. Currently this process is subjective in nature as we have no algorithmic means with which to identify or measure the usefulness of contextual attributes. In identifying individual attributes, the key is to find the most uniquely discernable characteristics

of their cyber behavior. Attributes spanning a range of activities will also aid in focusing the individual profile. Sampling characteristics associated with occupation, hobbies, alma mater, music interests, etc. statistically narrows the population of those meeting the profile criteria.

Group characteristics can be somewhat easier to identify, but their effectiveness is highly dependent on the scope of interests in the group returned. For example, one may define “programming” as an attribute of interest for a given group, however on instantiating the profile find a substantial portion of the population exhibit this characteristic. Refinement often focuses a profile enough to achieve meaningful differentials, otherwise the attribute will have to be abandoned for one more specific.

Steps two and three involve querying our activity ontology labels and descriptions in order to identify activities used to describe the attributes from step one. As outlined in Section 4.1, our activity ontology is a hierarchical labeling structure consisting of concise descriptions associated with each label. While the labels themselves may provide a direct mapping from attribute to activity (i.e. *Computers/Programming* to describe computer programmers), less obvious activities are identifiable by querying the label descriptions. For a user working in the area of game theory, the combination of the terms “game” and “theory” are obviously best suited to describe this attribute. A keyword search of category names reveals *Science/Social Sciences/Economics/Game Theory* and *Science/Math/Combinatorics/Combinatorial Game Theory* as the top two results. However, by searching activity descriptions, we are able to discover *Games/Game Studies*, *Computers/Artificial Intelligence/Games*, and *Science/Math/Logic and Foundations/Game Semantics* as additional activities related to this attribute.

Step four is to take the activities identified in steps two and three and examine “See also” activities within the ontology. Beneath the hierarchical labeling scheme of our ontology, there exists the “See also” link structure connecting “related” categories. These relationships

are human identified and done so in accordance with strict guidelines [153]. Like querying descriptions, “See also” categories aid in identifying contextually related but potentially non-intuitive activities (i.e. a “See also” category for *Computers/Artificial Intelligence/Games* is *Computers/Programming/Games*). Again, depending on the scope of the profile and other interests of the user, this could be a highly relevant category. As with step one of this process, steps two through four currently require manual examination and consideration in choosing the most appropriate activity descriptions. One method to address this is through the use of social network analysis and link discovery methods. The underlying “See also” structure of our ontology connects the preponderance of our activity nodes thus creating a very large and complex directed graph. By querying our ontology and selecting all categories returned, we are effectively extracting an attribute specific sub-graph. Using methods such as entropy models [204] to identify the most interesting and important nodes within this subgraph allows for more quantitative selection of key activities. While algorithmic processes and procedures might simplify this process, we currently foresee no truly automated mechanism to do so.

The instantiation step takes the activities discovered in steps two through four and identifies the most representative user or users meeting these profile criteria. Identification is done using a variety of similarity measures (see Section 5.4.3) and is dependent on the objective of the analysis. If searching for a group of individuals who have taken part in some activity, a 0^{th} order measure (see Section 5.4.3.1) is best suited. However, if the goal is to identify an individual based on the number and type of activities performed previously, a 1^{st} order measure (see Section 5.4.3.2) would be more appropriate.

Because of the breadth of activity labels, “wildcarding” is used to specify activities containing certain words and to group lower level activities. Used in databases or regular expressions, specifying a wildcard (normally by an asterisk “*” character) is equivalent to matching any zero or more characters. The string “wildcard*” would match “wildcards”, “wildcard birth”, “wildcard matching is fun”, etc. Using our game theory example, we

would be able to wildcard the terms “game theory” as “*game theory*” in order to match all activities containing those terms. We can also wildcard higher level activities such as “Computers/Programming*” to identify all lower level computer programming activities.

The output of the previous step is one or more users meeting the profile characteristics. The final step of contextual profiling is to determine if these users are representative of the profile or if further refinement is needed. As stated throughout this section, profile definition is a very qualitative process and one often performed in an iterative manner. Activities identified and instantiated in the previous steps may provide results which are too broad or narrow in focus to be useful. Analysis of the results to determine which activities are overly general as opposed to those which may be overly specific is a key step in the profiling process. Profile characteristics often have to be fine tuned, added, deleted, or modified a number of times before becoming sufficiently descriptive of the individual or group in question.

5.4.1.2 Temporal Attributes

Unlike contextual attributes which describe *what* a person does, temporal attributes describe *when* and *how often* certain activities occur. Like their contextual counterparts, temporal attributes can be specified at varying levels of fidelity. If interested in personnel working the late shift, a temporal attribute may be users online between 2300 and 0700 hours. In addition to the *when* component of temporal attributes, we also use *how often* as a distinguishing profile parameter. This attribute is often associated with the level of interest someone has in an activity and is determined by monitoring for transience or persistence (see Section 5.5.1.1) in one’s browsing of certain categories. Someone with a persistent interest in sports might check scores and news every day, while an individual with a transient or fleeting interest may do so only once. Using techniques in Section 5.5.1.1, these characteristics can be both identified and labeled, providing an additional tool to characterize individuals or groups.

Temporal attributes are also used to enable and enhance contextual ones (and vice versa).

An employer wanting to identify employees reading online news when they first arrive at work and during the last hour of the day can use the *News* activity to filter on that topic. Assuming a normal distribution of *News* related activity around the start and end of the day, a 1st order temporal profile will accurately identify these personnel. Figure 5.10 is the resultant data capture of a user fitting this profile. As with contextual attributes, there is

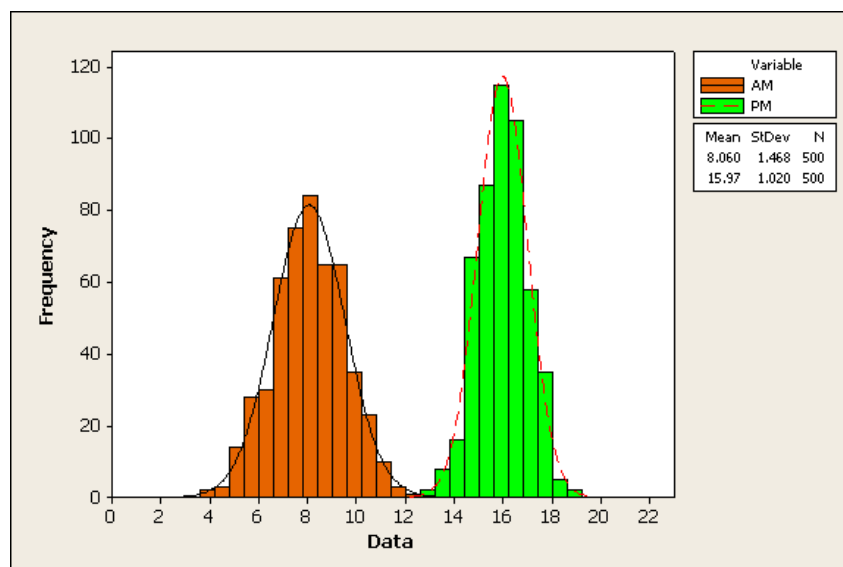


Figure 5.10: User profile making use of temporal and contextual attributes to identify those browsing *News* in the AM and PM hours

no quantitative mechanism to identify or rank temporal attributes.

5.4.2 Behavioral Fingerprints

While individuals prefer to think of themselves as free-willed, self-governing, and unique, research suggests the exact opposite may be true. Work done by Eagle et al [64] tracking the geographic location of college students found approximately ninety percent of what most people do in any day follows routines so complete their behavior can be predicted with alarming accuracy. Similar research [43][49][86] indicates a “creature of habit” mentality also

occurs in our patterns of Internet use. While e-commerce has taken advantage of browsing patterns for years for personalization and recommender systems, little to no work exists to suggest if these same patterns are descriptive enough to uniquely identify an individual user. In the remainder of this section, we demonstrate how a behavioral fingerprint is created and used to distinguish someone based on 1st order behavioral traits.

Although fingerprinting is broken out in its own section, the identification and detection of an individual with these techniques provides an ideal profile characteristic (see Section 5.4.1). Using a combination of contextual and temporal attributes a behavioral fingerprint can function either as part of an individual or group level profile.

Though a number of approaches exist to define user fingerprints from online activities, research done in document retrieval from the Information Retrieval (IR) domain seems to provide the most appropriate fit. Similar to Section 5.3, a user is described by a probability distribution over k possible activities with probabilities p_1, \dots, p_k . Frequency is calculated as the number of unique activities per session over all sessions. Our goal in doing this is to build a “user” search engine where given a sample of user activities, we can query all stored users to find the one most representative of the input. By defining random variable X_i as the number of times activity i is uniquely observed per session over n sessions, we represent a user by the activity vector $U = (X_1, \dots, X_k)$. With this in mind, given some initial training data $(x_1, u_1), \dots, (x_n, u_n)$, we produce a classifier $h : X \rightarrow U$ mapping a user activity vector, $x \in X$, to the user who most likely generated it, $u \in U$, as defined by some learned ground truth mapping $g : X \rightarrow U$.

5.4.2.1 Fingerprint Accuracy

To test our fingerprinting approach, we used the America Online (AOL) data set. Released in August of 2006, this data consists of three months worth of search queries (approximately 20 million queries) by 657,426 AOL subscribers (Section 6.1.2 provides a detailed description

of the data set). To scope the problem, we randomly selected 10,000 users from the data set, sessionized the queries, and extracted activities for each query/click link combo. Based on data from [136][118][23], we estimate the average number of browsing sessions per user per month to be approximately 30 and the average number of query based sessions per user per month to be 20 for 2006. To minimize the noise introduced by very low query counts, we eliminated users from the data set exhibiting less than “average” query activity for this time period (users with less than 20 sessions per month) and select users with sixty or more sessions over the three month period. After filtering, our sample of users shrinks from 10,000 to 1,092. In order to mimic real world usage, temporal order of the data is maintained (i.e. for an 80/20 train test split, the first 80% of the data is used to train the fingerprint and the remaining 20% represents a new sample generated by the user).

We initially use a vector space model (VSM) as the basis for identifying users as it is perhaps the best known and most widely used model in IR for document retrieval systems [120]. The vector space model maps user activities to user vectors in n -dimensional space. A new vector (query vector) is then compared to the user vectors using a predefined similarity measure, and a relevance ordered list of users is returned. We make use of activity frequency, inverse user frequency (same concept as term frequency inverse document frequency) for activity weighting, and Jaccard distance for the similarity measure.

We evaluate fingerprint accuracy using both a naïve sample size estimation (Section 5.3.1) and our empirical sample size approach (Section 5.3.2). For each sample size estimation method, the 1,092 users are partitioned into ten groups of one hundred. Non-stratified (temporal order is maintained) 10 fold cross validation is performed on increments of ten users for each group and the average result for all ten groups is calculated and recorded. Fingerprinting is performed using the click-link URLs, the full activity description of each query, as well as the top four, three, and two levels of the activity description (i.e. activities represented as A/B/C/D/E are also evaluated as A/B/C/D, A/B/C, and A/B). Figure

5.11 represents the number of dimensions required to describe all 1,092 users for URLs and various activity levels. Using full activity descriptions, we see a 33% reduction in the number

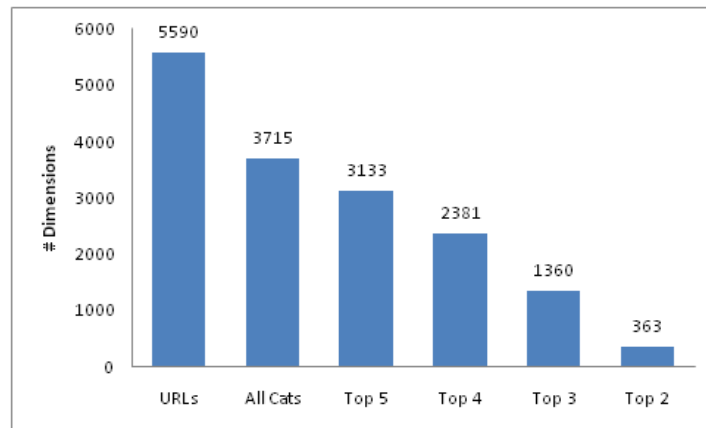


Figure 5.11: Number of dimensions required to represent 1,092 users with URLs and various levels of activity labels.

of variables needed to identify an individual and a 94% reduction if using only the top two levels of the activity label.

To determine the naïve sample size for the population, we start with 100% of the data and iteratively remove 10% of the queries from each user and evaluate using 10 fold cross validation. The results are shown in Figure 5.12. While not overly surprising, the highest accuracy (87%) was obtained by using all of the data. Because some data is needed to test, we use a 90/10 train test split. In the context of fingerprinting, this is equivalent to creating fingerprints of all users based on 90% of their generated traffic and then trying to identify the user responsible for a 10% sample. We evaluated the data using this criteria in ten person increments on the ten one hundred person subsets described above. The averaged results of this are presented in Figure 5.13. Even using a naïve selection criteria, we see strong indications that user fingerprints exist and are capable of identifying individuals with relatively high accuracy, even when using just the top 2 levels of activity labels.

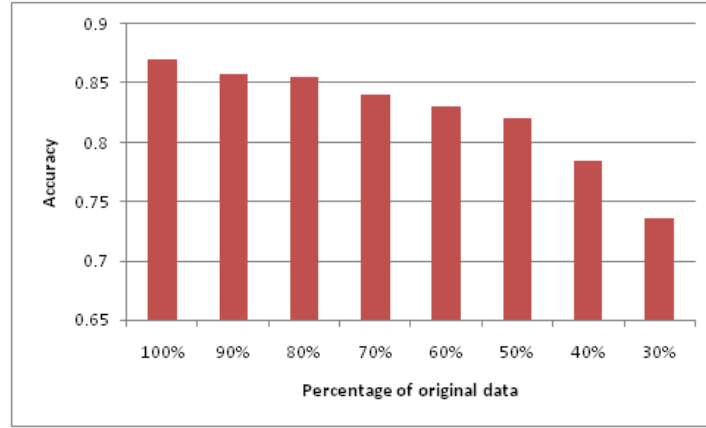


Figure 5.12: Determination of naïve sample size by iteratively removing 10% of data and evaluating classifier accuracy using 10 fold cross validation. Using 100% of the data yields the highest (87%) accuracy.

The experiment was re-run using our empirical sample size technique. Sample sizes were determined based on the amount of data needed by each user so that each multinomial parameter was within 10% of the true value with a 95% credible interval. While variable amounts of data are required for each user, our empirical approach on average required 34% less data per hundred user subset than using the naïve method. Figure 5.14 shows the accuracy using empirical sample size estimates on the ten one hundred person subsets.

In addition to requiring less data, these results show the fingerprinting accuracy using the empirical approach is overall better than using the naïve approach. Examination of the naïve average when using all activity labeling information (91.7% accuracy), shows this to be within 2% of using the empirical average when using the top three labeling levels (89.7% accuracy). Using the empirical sample size calculation yields a data reduction of 34% and a dimension reduction of 63% while still staying within 2% accuracy of using all of the data to fingerprint users. It should be noted that while using the click link URLs provided very high fingerprint accuracy for both the naïve and empirical results, we believe the large number of dimensions required to represent users with this data make it infeasible for larger populations.

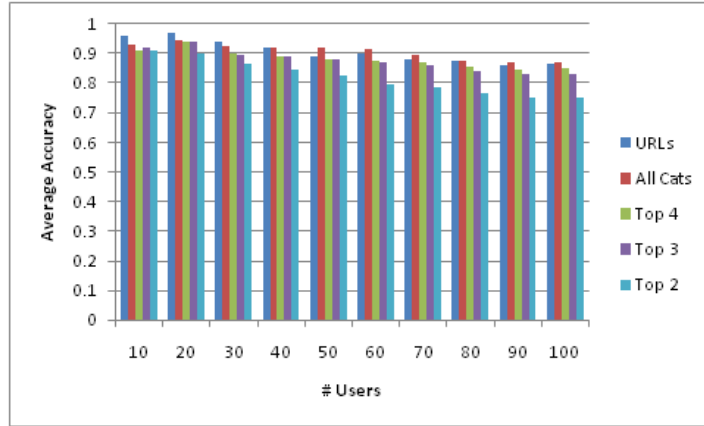


Figure 5.13: Evaluation of fingerprinting technique using a 90/10 train/test split on click-link URLs and various levels of activity descriptions.

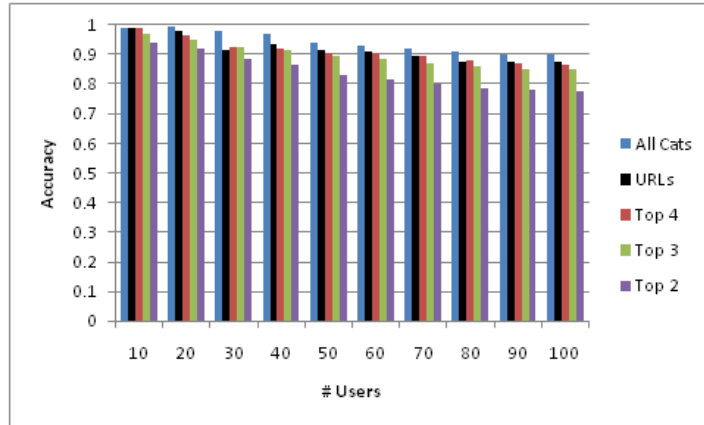


Figure 5.14: Evaluation of fingerprinting technique using empirical samples size estimates on click-link URLs and various levels of activity descriptions.

Though the VSM is a well known and understood model in the IR domain, other approaches have been shown to yield better results. To take advantage of the multinomial nature of our data, we perform one last experiment using a multinomial naïve Bayes classifier [133].

Naïve Bayes is a probabilistic classifier based on applying Bayes' theorem with strong independence (naïve) assumptions. Although a simple algorithm, naïve Bayes has been shown to perform extremely well in text classification tasks [195][133] compared to more

complex approaches. Multinomial naïve Bayes [105] extends this approach so that it may be used with multinomial variables.

Using Weka’s [224] multinomial naïve Bayes classifier, we re-evaluate our fingerprinting technique using empirical samples size estimates for the top 3 activity labels. One hundred person increments were evaluated instead of the ten person increases used in the previous experiments due to the very high classification accuracy using these intervals. Figure 5.15 displays the results of the experiment. Using this classifier we were able to classify three

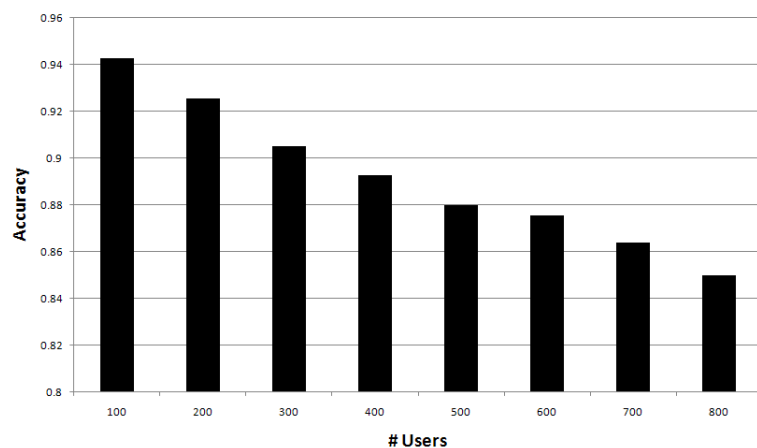


Figure 5.15: Evaluation of fingerprinting technique using empirical samples size estimates and multinomial naïve Bayes classifier on top three activity labels.

times as many users with the same accuracy as using a VSM classifier and were still well above 80% accuracy when analyzing eight hundred users.

Further research is needed in this area to determine an optimal classifier as well as credible interval ranges to maximize classification accuracy.

5.4.3 Cluster Analysis

While profiles are an excellent behavioral analysis tool to identify individuals or groups exhibiting certain behavioral traits, there are times when one will need to examine a group

of individuals in which no discernable attributes are known. Clustering provides a means to group “like” individuals based on various similarity measures. Unlike group profiles in which characteristics of interest are pre-defined, clustering identifies groups based on shared attributes which may or may not be known in advance. This type of analysis provides a “lay of the land” needed to determine the most prevalent characteristics within a group which may then be used to customize individual profiles.

The goal behind clustering is to characterize groups of individuals in a manner such that intra-cluster similarity is maximized and inter-cluster similarity is minimized. A cluster is therefore a collection of objects which are “similar” among them and are “dissimilar” to the objects belonging to other clusters. In this research, the goal is to utilize clustering to group together users exhibiting similar activities. Similarities can be based on *what* cyber activities are performed, *how often* they perform them, or the *order* in which they are performed. The remainder of this section outlines the similarity measures used to perform our user-based clustering.

5.4.3.1 0th Order Similarity Measure

The first similarity measure of interest is based on *what* a user does. This measure simply determines the amount of overlap in activity labels between users without taking into account the frequency of the label. For example, user one may have single visits to *Sports*, *Science*, and *Computer* based web sites. User two may have ten visits to a *Sports* site and one visit each to a *Science* and *Computer* site. Using this similarity measure, these two users would be identical.

In terms of this measure, *usage* is defined as follows:

Given m users $U = u_1, u_2, \dots, u_m$ who performed n distinct activities $A = a_1, a_2, \dots, a_n$ in some time interval. For each activity a_i and each user u_j , we define a usage value $use(a_i, u_j)$ as:

$$use(a_i, u_j) = \begin{cases} 1 & \text{If } a_i \text{ is accessed by } u_j \\ 0 & \text{Otherwise} \end{cases}$$

From there, we then define the usage based similarity measure as follows:

$$UB_Sim(u_i, u_j) = \frac{\sum_k (use(a_k, u_i) * use(a_k, u_j))}{\sqrt{\sum_k use(a_k, u_i) * \sum_k use(a_k, u_j)}}$$

This measure may be used as an initial step in analyzing a group of users with unknown behavioral characteristics. Because frequencies are not taken into account, little emphasis should be placed on the users identified in these clusters, but rather co-occurrences of activities within clusters.

5.4.3.2 1st Order Similarity Measure

While a 0th order similarity measure is useful for an initial view of the environment, as seen in the previous example, it fails to acknowledge the level of interest of the users. A 1st order similarity measure addresses this *how often* aspect of behavior by tracking the number of times users access common data types. This measure is defined as follows:

$$FB_Sim(u_i, u_j) = \frac{\sum_k (acc(a_k, u_i) * acc(a_k, u_j))}{\sqrt{\sum_k (acc(a_k, u_i))^2 * \sum_k (acc(a_k, u_j))^2}}$$

where $acc(a_k, u_i)$ is the total number of times that user u_i accesses the activity a_k .

Unlike the previous measure, this one separates users who perform an activity once a day versus a user performs the same activity many times a day.

5.4.3.3 Viewing Time Based Similarity Measure

In addition to frequency of activities, user's interest can also be measured by monitoring the length of time spent performing an activity. A user who spends a disproportionate amount of time viewing information on a specific topic may provide insight into how important or

interesting the topic is to the user. This metric also provides a means to identify a user's familiarity with a given resource. For instance, a user might not remember the full URL for a website of interest, but can recall the main domain name. This user goes to the site of interest on a daily basis by entering the primary domain name, then immediately clicks on the sub-link of interest. A user unfamiliar with the primary domain of the same site will most likely spend additional time reading the content and becoming familiar with the navigational options before selecting the link leading to the same sub-link within the domain.

Let $t(a_k, u_j)$ be the time the user u_j spends viewing content related to activity a_k . The similarity between users can then be expressed as follows:

$$VT_Sim(u_i, u_j) = \frac{\sum_k (t(a_k, u_i) * t(a_k, u_j))}{\sqrt{\sum_k (t(a_k, u_i))^2 * \sum_k (t(a_k, u_j))^2}}$$

While this measure offers excellent insight into individuals level of interest in an activity, the ability to accurately measure and collect data to correctly categorize behavior is much more of an art than a science. Determining whether someone is “actively” reading a file versus just opening the file and heading out to lunch is a level of clarity achieved only with a complex monitoring system. While not used in this work, this measure is included if a viable data capture mechanism becomes available.

5.4.3.4 Visiting Order Based Similarity

Perhaps the most rigorous of the similarity metrics discussed thus far, visiting order based similarity measures the frequency and the order in which activities were performed. This measure combines aspects of *what*, *how often*, and *in what order* activities are performed.

Using this measure, two users are identical only if they perform a sequence of activities in the exact same order and with the exact same frequency. Research in this area [82][21] suggests Levenshtein Distance and Sequence Alignment Methods to be appropriate means to calculate this metric. We use Levenshtein distance in our research.

Levenshtein distance is a metric used in information theory and computer science to measure the amount of work required to transform one sequence into another. Commonly used by spell checkers, the metric tracks and weighs operations (insertion, deletion, or substitution) needed to transform one stream into another.

We define visiting order based similarity between two activity sequences s_1 and s_2 as follows:

$$d(s_1, s_2) = (w_d D + w_i I) + nR$$

where d is the distance between two sequences s_1 and s_2

w_d is the weight value for the deletion operation ($w_d > 0$)

w_i is the weight value for the insertion operation ($w_i > 0$)

D is the number of deletion operations

I is the number of insertion operations

R is the number of reordering operations

n is the reordering weight ($n > 0$)

All of the similarity measures defined have been implemented in Java for this project. We use the CLUTO [99] software package for the clustering of the data based on the similarity measures just defined. CLUTO allows the clustering of low and high dimensional data sets using three classes of clustering algorithms (partitional, agglomerative, and graph partitioning). A complete analysis of these clustering algorithms is beyond the scope of this document and additional details are available at [100]. While many open source and commercial clustering implementations exist, most restrict users to a number of commonly used similarity metrics (Euclidian distance, Jaccard coefficient, Mahalanobis distance, etc.). While CLUTO also has set of predefined similarity measures, it also contains a mechanism

to cluster based on the similarity space between objects. Therefore, CLUTO can be passed a similarity matrix computed from any similarity measure and compute clusters from it. Because of this, CLUTO offers a tremendous amount of flexibility and is ideally suited for this research.

5.4.4 Demographics

Demographics is the study of characteristics of human populations and population segments, especially when used to identify consumer markets. Commonly-used demographics include race, age, income, educational level, home ownership, employment status, and even location. Demographic data is used extensively by marketing researchers to determine what segments or subgroups exist in the overall population and to create a clear and complete picture of the characteristics of a typical member of each of these segments.

Previously, demographic data was compiled using information from sources including surveys, grocery store cards, and contest applications. Today, the preponderance of this data is collected from various online surveys, user account profiles, and various software and files placed on clients local machines (i.e. cookies, spyware, etc.). For the purposes of this research, we use Microsoft's adCenter Labs [140] and Quantcast [173] to obtain demographic information from URL and query data.

Microsoft's adCenter Labs is a Microsoft group focused on researching digital advertising technologies to more effectively connect advertisers and consumers [140]. To facilitate this process, the Labs have designed and made available a number of tools in areas ranging from Audience Intelligence to Keyword Research. One of the specific tools offered is for Demographics Prediction. This tool uses an individual's search queries and Web page views to predict age and gender. Results are calculated using adCenters predictive model generated by analysis of MSN Search log data. Both a General Distribution (from a one-month MSN

Search log) and a Predicted Distribution are given. The tool works with either keywords or URLs as input. By entering the most prevalent URLs and queries of a user into this tool, a probabilistic determination can be made as to an individual's age and gender. While we have not performed extensive testing in this area, using ground truth data captures for twelve individuals (see Section 6.1.1 for data details), we were able to accurately predict age ranges 83% of the time and gender 92% of the time.

Quantcast is a web analytics company designed to provide publishers and marketers information on Internet audiences ranging from demographics to lifestyle preferences (i.e. sites also liked by audience, types of search queries made by audience, etc.). Figure 5.16 displays the type of demographic information which may be obtained for the URL www.dartmouth.edu. Unlike Microsoft's adCenter which allows this type of information to be queried by URL or

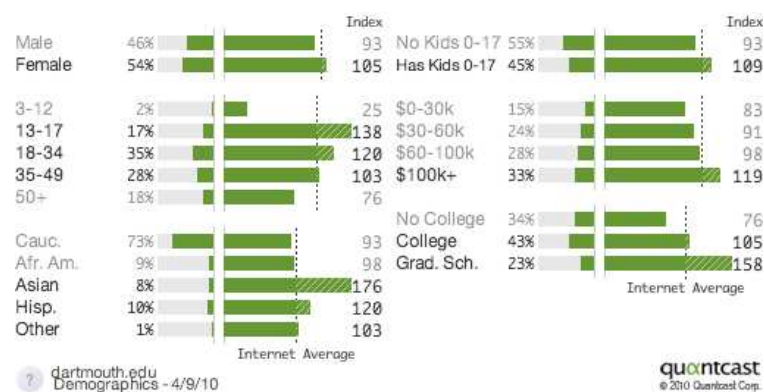


Figure 5.16: Demographic information returned from Quantcast for www.dartmouth.edu.

keyword, the Quantcast site only allows URL based searches. An additional service offered by the site is the ability to define demographic profiles (age, race, gender, etc.) and generate a list of URLs matching the profile characteristics (a relevance score to the profile is also given). An area of future research involves using this service to determine if demographics can be generalized for categorical information such as that used in our activity ontology.

While it is understood demographic profiling is a probabilistic generalization of a group,

when combined with additional cyber and non-cyber traits, this information is a valuable resource for accurately characterizing an individual.

5.5 Prediction

The ability to predict future cyber behaviors and actions from previously observed activities is an important aspect of behavioral modeling. While a fair amount of literature exists on cyber-based prediction (see Section 2.1.2.3), most of this work leverages a small set of probabilistic models to foresee future activities. In the remainder of this section, we discuss a number of these algorithms and how they are used with our methodology, but also define a new means to characterize and predict online behavioral trends using the concept of *behavioral state*.

5.5.1 Behavioral State

While a number of definitions of *state* exist, we define the *behavioral state* of an individual to be the characterization of the length and frequency of interest in a cyber-based activity or behavioral trait. We further classify these states as being either transient or persistent in nature.

A *persistent behavioral state* (PBS) is characterized by an activity or behavioral trait demonstrated consistently for an extended period of time or repeated in regular intervals. *Extended period* is the amount of time required to adequately characterize an individual or group being monitored. This is a period of time greater than or equal to the average sliding window size as defined in Section 5.3.2. *Consistency* is determined as the behavioral steady state where the proportional frequency of the behavior or trait is unchanging over time. If a user is behaving in a consistent manner, then the recently observed behavior of the user will continue into the future.

There are numerous reasons a user may be exhibiting PBS. A behavior defined by the lack of a definitive goal or definable time frame determining completion is one cause of a PBS. Someone thinking about buying a house may fall in this category. Years of research may be conducted on real estate, locations, mortgage rates, etc., but until a certain circumstance presents itself, no closure may ever occur. Another reason for being in a PBS is simply a product of the difficulty or amount of work associated with it [198][113]. Getting a PhD is a daunting task requiring years of focused research in a number of well defined areas. This will present itself in the cyber realm through consistent browsing in a finite number of fields. A final reason for PBS is simply attributed to monitoring for changes and updates to information [113][102]. This type of PBS is normally attributed to a hobby or long term interest of a user. An individual interested in sports is going to continually monitor the progress of a favorite team over an indefinite period of time.

Transient behavioral state (TBS) is associated with those behaviors persisting for a succinct period of time and having a specific goal or end state. *Succinct* here refers to a period less than the average sliding window size as defined in Section 5.3.2. A user exhibiting a transient behavioral state will perform activities until a goal is met, but the time period is not large enough to be considered persistent. A TBS may be caused by a significant event in one's life or a recent news story. For example, a user not normally interested in Michael Jackson may visit a large number of web sites and conducts numerous searches related to the singer upon hearing of his death. Following the initial surge, no further interest in the topic exists and cyber related activities drop to normal levels. A TBS can occur in a single session or persist over multiple browsing sessions. A user looking to buy a car generally spends numerous sessions comparing prices and reading reviews until the goal of buying a car is achieved, at which time interest in car related activities will subside. It should be noted a PBS will initially show up as a TBS until enough data has been collected for the *extended period* criteria to be met.

5.5.1.1 State Labeling

Transient and non-periodic persistent behavioral states are both identifiable in online data by monitoring unique activities per session. We are interested in the proportion of unique activities per session vice total activities per session to depict persistent interest over time rather than a very focused, fleeting interest during a given period. Tracking the proportion, vice actual counts, also provides a more stable representation of behavior. For example, activity *A1* has the following total counts for sessions *S1* through *S6*:

S1 - 10
S2 - 0
S3 - 3
S4 - 1
S5 - 95
S6 - 0

Figure 5.17 is a plot of total counts per session (left) and proportion of sessions over all sessions (right). Even with a relatively small sample size, this figure demonstrates how total

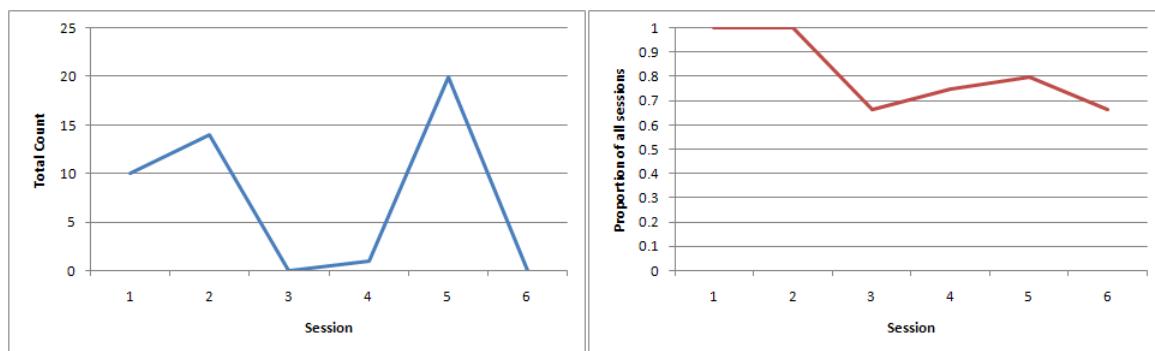


Figure 5.17: Plot of total counts per session (left) and proportion of sessions per session count data can be dominated by single session fluctuations.

Next, piecewise linear representation methods are performed to extract behavioral state information from session-based time series data. While several methods exist to abstract high

level representations of a time series to include Fourier analysis and wavelets, we elected to use piecewise linear representations in our work as the segments identified are most representative of our state-based approach. Piecewise linear representation is associated with the approximation of a time series T of length n by K straight lines.

A number of approaches exist [104][103] for transforming a time series into a piecewise linear representation. The Sliding Window algorithm was selected for its simplicity, intuitiveness and the fact that it can be used in an online (for real time processing) or offline (for batch processing) manner. The Sliding Window algorithm works by anchoring the left point of a potential segment at the first data point of a time series and then attempts to approximate the data to the right over increasingly longer segments. When the error at some point i exceeds a pre-specified threshold, the subsequence from the anchor to $i - 1$ is transformed into a linear segment. The anchor is moved to location i , and the process repeats until the entire time series is transformed into a piecewise linear approximation. Pseudocode for the algorithm is shown below [104].

```

Algorithm Seg_TS = Sliding_Window(T , max_error)
anchor = 1;
while not finished segmenting time series
    i = 2;
    while calculate_error(T[anchor: anchor + i ]) < max_error
        i = i + 1;
    end;
    Seg_TS = concat(Seg_TS, create_segment(T[anchor: anchor + (i-1)]));
    anchor = anchor + i;
end;

```

Each linear segment is then classified as being a PBS or TBS depending on the constraints outlined in Section 5.5.1. Further state delineation from the slope of the segment is made as follows:

1. State P1 : Persistent Flat – persistent linear segment having an angle of incline θ such that $-3^\circ \leq \theta \leq 3^\circ$
2. State P2 : Persistent Increasing – persistent linear segment having an angle of incline θ such that $\theta > 3^\circ$
3. State P3 : Persistent Decreasing – persistent linear segment having an angle of incline θ such that $\theta < -3^\circ$
4. State T1 : Transient Flat – transient linear segment having an angle of incline θ such that $-3^\circ \leq \theta \leq 3^\circ$
5. State T2 : Transient Increasing – transient segment having an angle of incline θ such that $\theta > 3^\circ$
6. State T3 : Transient Decreasing – transient linear segment having an angle of incline θ such that $\theta < -3^\circ$

The three degree cutoff is a qualitative measure decided on after performing the experiments of Chapter 6. Empirical analysis of one's data source combined with the type of analysis to be performed is necessary to effectively calculate this value and determine if additional states are required (i.e. $3^\circ \leq \theta \leq 45^\circ$, $\theta > 45^\circ$, etc.). Once states have been identified and labeled, *inter-state* and *intra-state* predictions can be made.

Inter-state predictions take advantage of the characteristics of persistent states to ascertain future activities. The black line in Figure 5.18 is a plot of the proportion of unique activities per session. The initial window size is calculated to be 101 and has an average value of 98 by session 312. From session 102 to 312, the state of the user is determined to be in PBS P1 and is depicted on the figure as a dotted blue line. Upper and lower confidence bounds are calculated and added to the plot as dashed green and red lines respectively. Given the current state of the user, predicting future states is accomplished by merely extending our linear fit from the last observation forward using the same calculated slope value. The solid blue line beginning at session 312 and extending to session 347 shows the future predicted proportions for the activity in question. The solid purple line is the actual proportion of

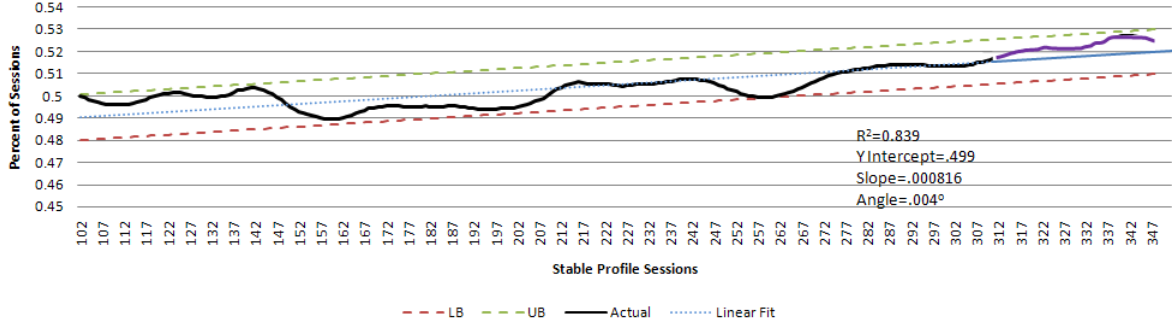


Figure 5.18: Inter-state prediction (solid blue line) thirty five sessions into the future with solid purple line depicting actual activity.

activities and is shown to stay within 0.1 of the predicted values for all thirty five sessions.

Fano's inequality correlates information entropy and data value predictability by defining the lower bound on the error probability of any data stream. Given a user with entropy S who has taken part in N sessions, we can determine their level of predictability by the following equation:

$$H(X|Y) < P_e(S, N)$$

given $H(X|Y)$ is the entropy of X conditional on Y and $S = H(P_e) + (1 - P_e)\log_2(N - 1)$. $H(P_e)$ is the binary entropy function described by $-P_e\log_2(P_e) - (1 - P_e)\log_2(1 - P_e)$. For a user with $P_e = 0.8$, at least 80% of the time the individual performs the behavior or activity in a predictable manner, and the remaining 20% of the time does so randomly. In other words, no matter how good our predictive algorithm, we cannot predict with better than 80% accuracy the future cyber activity of a user with $P_e = 0.8$. Therefore, P_e represents the fundamental limit for each individual's predictability. Figure 5.19 is a plot of P_e versus sessions for the graph of Figure 5.18. As the number of sessions in the same state increase, so does the predictability of future behaviors occurring in the same state. While after only thirty four sessions P_e is at a relatively high value of 0.8, the idea behind predicting based on persistence is that the behavior has demonstrated consistency. Another mechanism to determine when state transition will occur is through intra-state prediction.

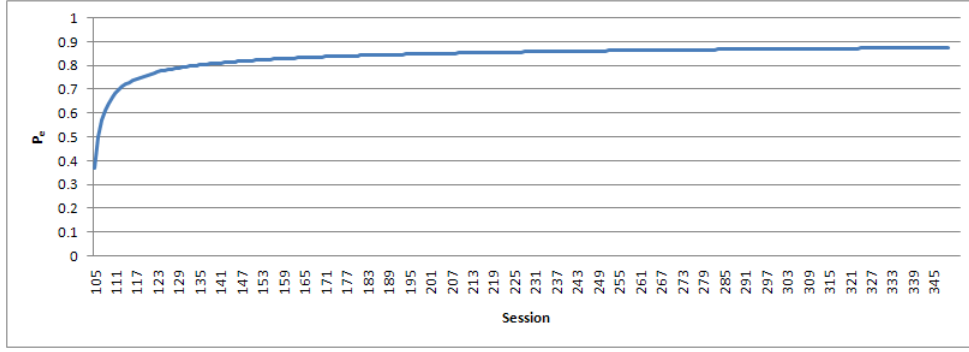


Figure 5.19: Plot of the predictability of the behavioral states of Figure 5.18. As the number of sessions in the state increase, so does the predictability of future behaviors occurring in the same state.

Intra-state predictions are modeled as 2^{nd} order Markov model describing the probability of transitioning between behavioral states. This model depicts the state of an individual's behaviors over time and provides insights in to the long term dynamic of the activity or behavior. For example, a user's behavioral states modeled as in the left hand side of Figure 5.20, demonstrate a persistent long term interest of the user allowing us to predict future activities in this area with high accuracy. The Markov model on the right hand side of

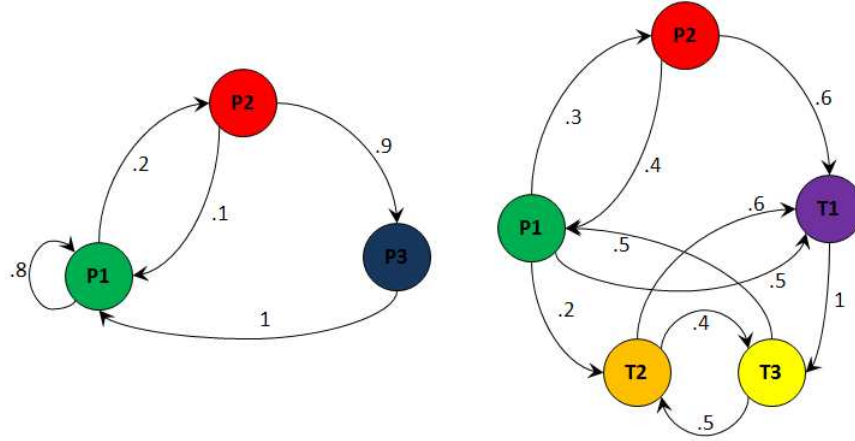


Figure 5.20: Markov model depicting user with a clearly persistent behavior (left) and a fairly unstable behavior (right) over time.

the figure is characterized by random spikes of possible increasing or decreasing transient

activity, making this model highly imprecise from a predictive standpoint.

We initialize the intra-state transition matrices with some a priori knowledge based on state type. An initial assumption is made that a persistent state will stay in a persistent state and a transient state will transition to any other transient state with equal probability. As additional observations occur, the transition probabilities are dynamically updated. Figure 5.21 shows the initial transition matrix prior to any observations. At any given time, we can

	P1	P2	P3	T1	T2	T3
P1	1	0	0	0	0	0
P2	0	1	0	0	0	0
P3	0	0	1	0	0	0
T1	0	0	0	0.333	0.333	0.333
T2	0	0	0	0.333	0.333	0.333
T3	0	0	0	0.333	0.333	0.333

Figure 5.21: Initial intra-state transition matrix prior to any observations. Transitions are based on the assumption that a persistent state will stay in a persistent state and a transient state will transition to any other transient state with equal probability.

calculate the long term or *steady state* vector representing the probability of being in a given state over all time, independent of the initial conditions. The steady state vector is defined as:

$$q = \lim_{n \rightarrow \infty} \mathbf{x}^{(n)}$$

This will converge to a strictly positive vector as long as the transition matrix P is regular (at least one P^n with all non-zero entries). Given the basic two state Markov model of Figure 5.22, we calculate the steady state values as follows for transition matrix P :

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{q}P = \mathbf{q}$$

$$= \mathbf{q}I$$

$$\mathbf{q}(P - I) = \mathbf{0}$$

$$\begin{bmatrix} q_1 & q_2 \end{bmatrix} \begin{bmatrix} -0.1 & 0.1 \\ 0.5 & -0.5 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$= \mathbf{q} \left(\begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)$$

$$= \mathbf{q} \begin{bmatrix} -0.1 & 0.1 \\ 0.5 & -0.5 \end{bmatrix}$$

From this we have $-0.1q_1 + 0.5q_2 = 0$ and because q_1 and q_2 are probability vectors, know $q_1 + q_2 = 1$. Solving the pair of simultaneous equations gives the steady state distribution:

$$\begin{bmatrix} q_1 & q_2 \end{bmatrix} = \begin{bmatrix} 0.833 & 0.167 \end{bmatrix}$$

We therefore conclude that 83% of the time the user will be in persistent state $P1$.

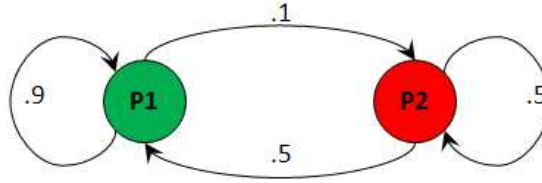


Figure 5.22: Simple two state Markov model used to demonstrate steady state behavioral calculations.

While our work in intra-state prediction is still in its preliminary stages, we believe the concept is sound and foundational to behavioral prediction. Additional research integrating work done in Markov processes [106] and embedded Markov chains should greatly enhance both intra and inter-state accuracies by integrating the amount of time (sessions) spent in a given state before a transition occurs.

5.5.2 Association Mining

As outlined in Section 2.1.2.2, the objective of association mining is to find all co-occurrence relationships within a given set of data. While commonly used in commerce to determine frequent combinations of purchased items, the goal within this research is to identify common trends in the data to predict future actions.

In general, the input to frequent item set mining and association rule induction is a number of transactions defined over a set of items. By treating *sessions* as transactions and *activities* as items, we use association mining algorithms and techniques to extract behavioral trends in both *online* and *offline* data.

While association mining analysis can be performed on individuals, it is most insightful when looking at the data set as a whole or clusters within the dataset. We currently use aspects of the Java data mining framework Weka [224] and a C implementation of the apriori, Eclat, and FP growth algorithms [32] to conduct this analysis.

5.6 Anomaly Detection

Anomaly detection refers to detecting patterns or trends in data not conforming to an established “normal” behavior. Trend analysis is a generic term referring to the concept of collecting information and attempting to spot a pattern, or trend, in the data. While trend analysis is often used to predict future events, we use these analysis techniques to first determine trends and then identify activities falling outside the trends.

5.6.1 CUSUM

Traditionally used to monitor manufacturing processes and signal anomalies in performance [142], Cumulative Summary (CUSUM) control charts have been making a come back of

sorts in recent years as an anomaly detection mechanism in the computer security realm [98][206][225][10]. We use CUSUM charts in this work to identify and detect positive and negative changes in a user’s behavior. We chose CUSUM over other anomaly detection schemes as this approach does not presume the nature of the change (linear, trend or otherwise) and treats positive and negative deviations equally. Additionally, a CUSUM monitoring scheme is very simple to implement, yet provides a formal and statistically sound framework to monitor the status of users behaviors.

In CUSUM charts a cumulative sum of the positive and negative deviations of the sample values from a target value is observed. Chart values are based on a set of observations x_i collected for time period $i = 1, \dots, m$ where their in-control mean μ and standard deviation σ are known. We currently estimate these values during sample size estimation (see Section 5.3) or by using standard statistical techniques [143]. Data are first normalized through the transformation $z_i = (x_i - \mu)/\sigma$ so deviations from μ are in units of σ .

The decision-interval CUSUM works by recursively accumulating positive and negative deviations separately with two statistics:

$$S_i^+ = \max[0, S_{i-1}^+ + z_i - k]$$

for positive deviations (“one-sided upper CUSUM”), and

$$S_i^- = \max[0, S_{i-1}^- + z_i + k]$$

for negative deviations (“one-sided lower CUSUM”), with starting values normally set as $S_0^+ = S_0^- = 0$. A CUSUM chart is obtained by plotting these statistics against i . If measurements tend to stay above the in-control mean, the upper CUSUM S^+ develops an upward trend, whereas the lower CUSUM S^- shows a downward trend if observations are consistently below the mean. The chart’s parameter k (usually called the reference value) relates to the size of the smallest shift in the level of z which one is wishing to detect quickly,

where deviations smaller than k are ignored. The decision rule is to declare an out-of-control state whenever S^+ exceeds the decision interval h or S^- falls below h . The values chosen for the parameters h and k (measured in standard deviation units) determine the performance of the CUSUM chart. It is perfectly acceptable to set different $h - k$ pairs for upper and lower CUSUM's if changes in one direction matter more than in the other.

Figure 5.23 is a sample CUSUM chart showing an “in-control” process up to time 20. The remaining 10 time units show the impact of a small increase ($\hat{\sigma}/2$) on the upper CUSUM.

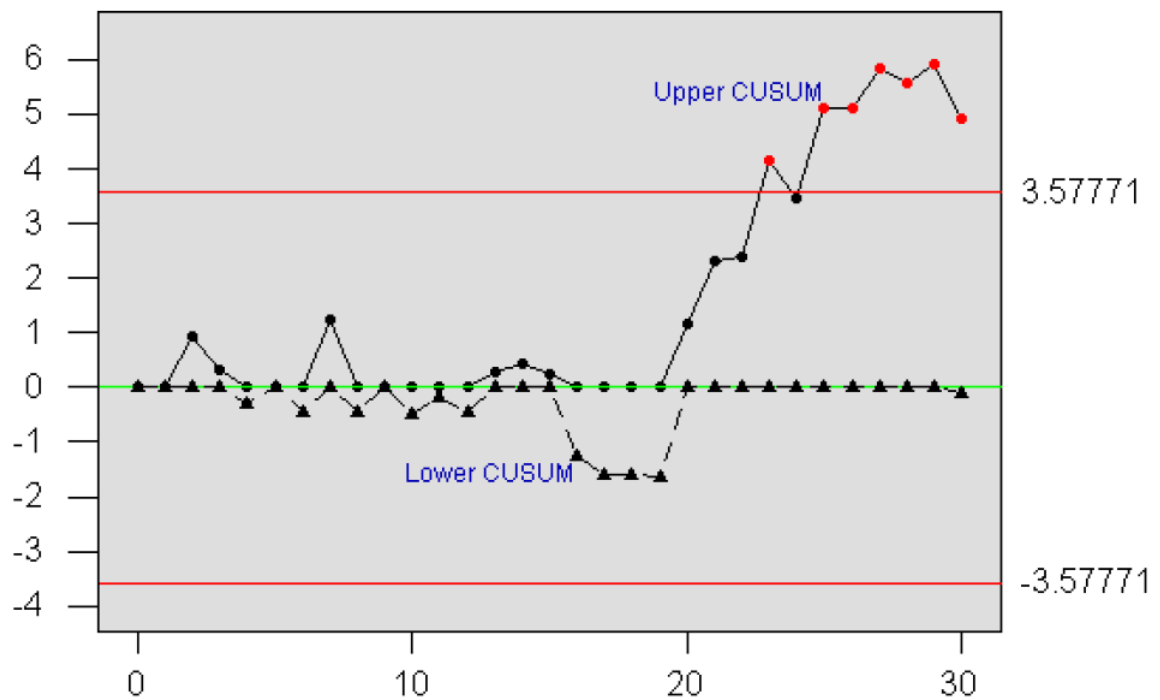


Figure 5.23: CUSUM chart showing an “in-control” process up to time 20. The remaining 10 time units show the impact of a small increase ($\hat{\sigma}/2$) on the upper CUSUM.

CUSUM control charts have been shown to be very effective in detecting small changes, but less effective in detecting large shifts [142]. Therefore, these charts are most appropriate for identifying changes in behavioral state (see Section 5.5.1) or percentage of sessions over time, but are not well suited to detecting large changes such as those found in queries per day

or queries per session. For detection of both large and small changes, we see our state-based change detection approach more appropriate.

5.6.2 State-Based Change Detection

While we have described how our state-based behavioral models of Section 5.5.1.1 can be used for prediction, they also provide a novel approach to behavioral anomaly detection. Using behavioral states allows for the detection of both general and specific behavioral changes. In the broad sense, one may wish to be notified anytime an individual's behavior or behavioral traits change state. Additional states may also be added in order to provide varying levels of fidelity on the types of changes occurring (i.e. add in a transient and persistent state detecting slope changes of forty five degrees or more). Someone wishing to effect the environment (i.e. e-commerce changing prices, a business blocking streaming media, etc.) may be interested in tracking all state deviations based on this change in order to analyze its effectiveness. Another general anomaly detection alert may be to monitor for changes from a persistent to transient state. Such signals may be for insider threat monitoring to notify management of sudden changes in one or more aspects of a user's behavior.

More specifically, one can monitor for certain sequences of behavioral states. Persistent flat ($P1$) behaviors followed by a transient increase ($T2$) and then decrease ($T3$) across an organization may be used to identify significant events taking place in a company (i.e. layoff announcements, short term suspense, etc.).

Work done in sequence learning for anomaly detection [112] and sequence databases search techniques [160][3] provide validation of both techniques using piecewise linear based data.

5.6.3 Fingerprint Deviation

As demonstrated in Section 5.4.2.1, behavioral fingerprints provide a highly accurate representation of a user based on cyber-based behavioral traits. In addition to providing a unique characterization of an individual, these fingerprints can also be used to monitor change. Kullback-Leibler (KL) information is a measure between conceptual reality, f , and an approximating model, g . KL information, denoted $I(f, g)$ is the information lost when model g is used to approximate reality, f . Using this measure provides a mechanism to monitor and plot changes in all activities over time. Once a fingerprint is established, new behavioral “snapshots” can be compared to this fingerprint in order to determine significant deviations over time.

5.7 Summary

In this chapter we have defined our behavioral model and outlined the methods used to analyze and interpret this model in a quantitative manner. In Chapter 6, we demonstrate how this model is instantiated and the methods utilized on real world cyber input.

Chapter 6

Empirical Evaluation of Cyber-Based Behavioral Models

In this chapter we evaluate the benefits of using cyber-based behavioral modeling in three disparate domains; military targeting, stress monitoring, and insider threat detection. Through use-case examples of each, we demonstrate both the depth and breadth of our approach by instantiating a diverse mixture of the analysis techniques of Chapter 5 at varying levels of detail.

6.1 Datasets

Due to the large amount of personal information inherent in a user's online and offline data, privacy concerns limit the amount of publicly available data sets to test our approach. With this in mind, the following two data sources are used in the testing and analysis of our methodology; browser history files and the America Online (AOL) data set. The first data set consists of twelve browser history files from consenting users ranging in duration from one to four months. These history files contain the majority (users would sometimes

use other computers or browsers during the period in question) of the user’s web browsing activity for those periods of time. Because the information was collected from known and consenting users, “ground truth” information is available for verification of all analysis and testing using this data set. Due to the limited population size of this data source, the AOL data set is also used in our testing. Released in August of 2006, this data consists of three months worth of search queries (approximately 20 million queries) by 657,426 AOL subscribers. While this data does not have the benefit of “ground truth” information, it is a valuable tool for testing many aspects of our behavioral modeling methodology. Due to the large amount of personal information inherent in user’s search terms (i.e. addresses, names, banking information, etc.), this data is not publicly supported or available from AOL, but is mirrored and downloadable from locations such as [70][78]. The next two sub-sections will provide additional details and statistics pertaining to each data set.

6.1.1 Browser History Files

As stated previously, this data was obtained by collecting browser history files from twelve consenting individuals. These individuals ranged in age from 25 to 72 and consisted of seven males and five females. A significant benefit of collecting browser history files is there is no need to pre-process and filter out all of the “garbage” requests (i.e. images, banners, ads, etc.) as outlined in Section 3.3.3. Only the specific URL visited and its time stamp are saved to the history file. Unlike the AOL data set, consisting solely of queries and the corresponding link followed, the browser history files collected contained both query and clickstream data. Having an adequate query-based data set in the AOL data, we focus our testing on pure clickstream information and ignore query traffic for the experiments conducted using this data.

For ease of processing, we only selected individuals using the Firefox web browser. Firefox

stores browsing history along with a great deal of other personal browsing information (i.e. most visited pages, number of time pages visited, recent bookmarks, etc.) in a number of SQLite database files. SQLite is a software library which utilizes a self-contained and serverless SQL database engine. We transferred a subset of this data into a MySQL database with the following fields:

- UserID - User ID number.
- URL - the full domain name of the URL visited
- Timestamp - the time at which the URL was visited

Using preprocessing and normalization techniques outlined in Section 3.3, we sessionize and label the URLs and add this information to our database under the following fields:

- session - unique session number
- category - normalized URL label

Static sessionization (Section 3.3.4) with an inactivity period of thirty minutes was used to identify the sessions. After sessionization we then mapped the URLs onto our ontology. Approximately 59% of the URLs were labeled via direct lookup in our ontology, 35% were classified using our Delicious classifier approach, and the remaining 6% of the links were omitted. Due to variability in the quality and amount of data present on any given web page, we decided to not classify those URLs requiring our tag extraction techniques (see Section 3.3.6) in order to minimize noise introduced by possible misclassifications. We were left with a data set consisting of 89,308 click links broken into 5,935 sessions with an average of 14.4 links per session.

Figure 6.1 is a 1st order model depicting the top level categories visited by the users in this data set (categories with less than one percent representation are omitted for readability purposes). In order for the data to not be skewed by individual users, we count the number

of unique visits per session (i.e. a user visiting a computer site 100 times during a given session is only counted as one visit for that category for that session) rather than simply sum the number of visits per category. Figure 6.2 is a listing of the top 10 full categories and

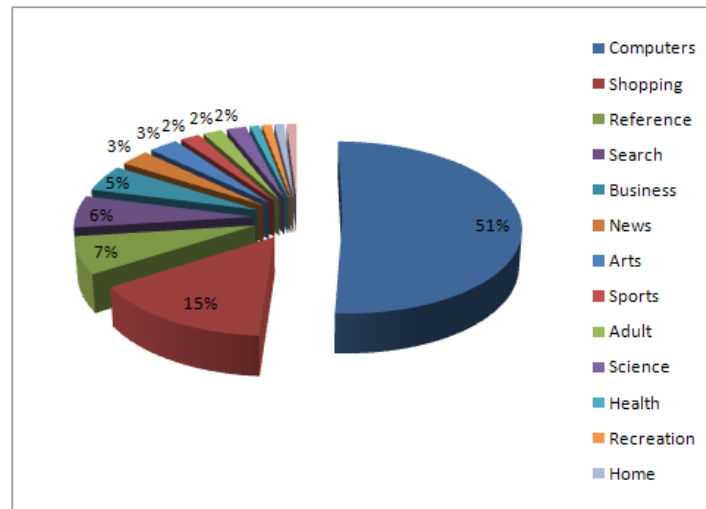


Figure 6.1: 1st order model depicting the top level categories visited by 12 users.

associated counts. Due to the small sample size, we will make no generalizations about the

Computers/Internet/E-mail/Web-Based	17861
Computers/Internet/On the Web/Online Communities	15182
Shopping/General Merchandise/Major Retailers	7207
Searching	5358
Shopping	4701
Reference/Education/Schools	3572
Reference/Encyclopedias	1786
Shopping/Auctions	1488
Recreation/Travel	893
Adult/Porn	446

Figure 6.2: Listing of the top ten full categories and associated counts for the “browser history” users.

population based on this information but will highlight its similarity to national averages.

Using data from the Online Publishers Association (OPA) [157] and Nielsen online, national

statistics in the areas of *Commerce* (online shopping), *Communications* (webmail, instant messenger, groups), *Community* (Facebook, MySpace), *Content* (sites providing news, information, and entertainment), and *Search* (Google search, Yahoo Search) are plotted in blue in Figure 6.3. By mapping the data in Figure 6.2 onto these higher level categories (i.e. *Online_Communities* \rightarrow *Community*, *Shopping* \rightarrow *Commerce*, etc), we are able to compare our users (plotted in red) against the national averages. While not an exact match,

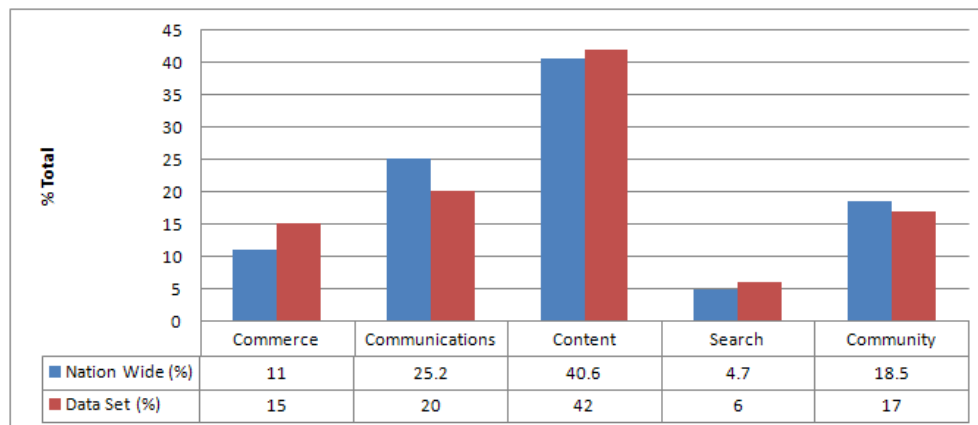


Figure 6.3: Comparison of national average browsing characteristics to our 12 “browser history” users.

this graphic does demonstrate our data is fairly representative of national online behaviors even with a relatively small sample size. Using our iterative sample size algorithms of Section 5.3, we are able to calculate with 95% confidence these samples to be within 6% of their true mean.

6.1.2 AOL Data

Unlike the browsing history data set containing both click stream and query data, the AOL data set is based on user query traffic only. While traditional query classification relies solely on the query, the AOL data has the added benefit of providing the URL of the website

followed based on the query results. This additional information provides great insight regarding query context which may be otherwise unavailable via the query alone. The AOL data set includes the following fields:

- AnonID - an anonymous user ID number.
- Query - the query issued by the user, case shifted with most punctuation removed.
- QueryTime - the time at which the query was submitted for search.
- ItemRank - if the user clicked on a search result, the rank of the item selected.
- ClickURL - if the user clicked on a search result, the domain portion of the URL in the clicked result.

In this research, we will attempt to delineate between an *entity* and AnonID as identified above. An AnonID, while representative of a single AOL account, may be used by one or more *entities*. An example of this may be a family of four who all share one computer. It is unlikely each family member will have an individual AOL account and will therefore all submit queries under the same AnonID.

Two types of query events are captured in the data set:

1. A user generating a query which returns no results of interest.
2. A user generating a query and then selecting an item returned in the result set.

In the first instance, data exists in the fields *AnonID*, *Query*, and *QueryTime*, but not *ItemRank* or *ClickURL*. Below is an example from the data set showing this type of event.

1268 gall stones 2006-05-11 02:12:51

The data represents user 1268 entering a query for *gall stones* on 11 May 2006, but not getting any results of interest. If the user requested the next “page” of results for a query, this appears as a subsequent identical query with a later time stamp. In the second type of query, data exists in all five columns because an item of interest was returned and followed by the user. An example of this occurrence is shown below.

This shows the same user initiating a similar search for *gallstones* on the same day and selecting the number one ranked result (<http://www.niddk.nih.gov>). If a user selects more than one link in the query result set, two events will be generated, but the timestamps, ItemRank, and ClickURL will differ. One last note of interest involves the ClickURL. As stated above, the ClickURL represents the domain only portion of the URL in the result set the user clicked on. Therefore, if a user were searching for *Dartmouth College* and http://en.wikipedia.org/wiki/Dartmouth_College was in the result set and chosen by the user, <http://en.wikipedia.org/> would be seen as the ClickURL.

Given the size of the AOL data set, initial experiments were conducted on the first 4,188 users who were responsible for 1,799,443 queries. This number was determined by using our sample size algorithm (Section 5.3) to determine the minimal number of sessions needed to be within 5% of the mean for a user having at least one session in each category. We calculated this to be 125 sessions and therefore selected all users having 125 or more distinct sessions. Like the browsing history data set, we stored the selected user data in a MySQL data base with the same fields as defined above (*userid*, *query*, *timestamp*, *itemrank*, and *clickURL*) along with fields for *session* and *category* as outline in Section 6.1.1.

The first step in query pre-processing requires the identification of the queries. Given that our data only consists of queries, this step has already been accomplished. The next step in the process, is spell checking. This was initially attempted using GNU Aspell [18], an open source spell checker, but was later abandoned for this experiment due to the large use of acronyms and common names in the queries. Further research in areas such as entity identification is needed to single out and ignore these types of words to facilitate spell checking of “common” terms. Stemming of queries was then conducted using a Java implementation of the Porter stemmer.

The last pre-processing step performed was sessionization. Although this data is not representative of a typical clickstream, it is in fact stream data requiring sessionization before it can be effectively analyzed. As with browsing history data, *static sessionization* is used with an inactivity period of thirty minutes. Sessions, as defined for this experiment, refer to active periods of searching rather than active periods of browsing. We are making some assumptions by defining sessions in this manner. A user may complete a search and find a relevant result, then enter this site and continue surfing without entering another query. Due to limitations of the AOL data capture, none of this additional browsing activity is captured. We therefore make the assumption a user will actively search until they find a relevant result.

Figure 6.4 represents a 1st order model of the top level categories for all 4,188 users selected. Similar to the browser history data set, we do not want our results skewed by individual users with abnormally high query counts in any one category and therefore again count the number of unique visits per session vice total counts per category. From the

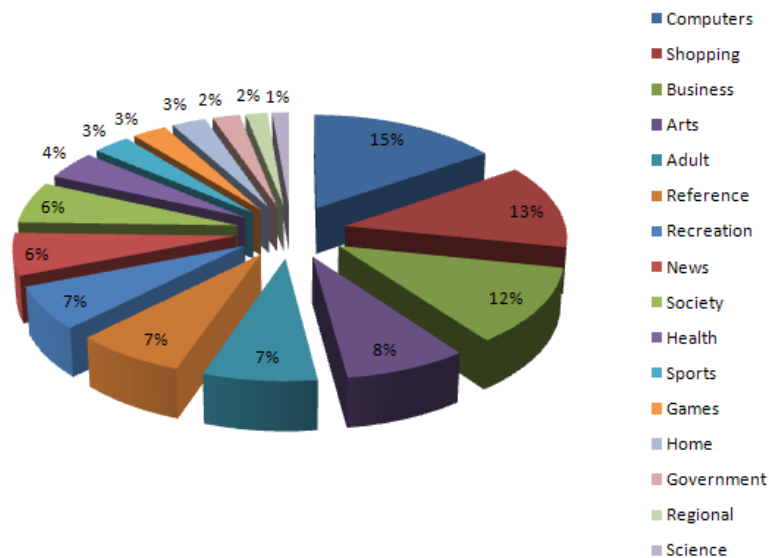


Figure 6.4: 1st order model of the top level categories for 4,188 selected AOL users.

graphic, the top three areas of interest are in *Computers*, *Shopping*, and *Business* followed by fairly balanced distributions in *Arts* through *Society* and *Health* through *Science*. Figure 6.5 is a listing of the top 20 full categories of interest to AOL users as well as their associated counts. Interestingly, this relatively small list of categories represents approximately 40% of all queries made by the data set users but falls into only ten top level categories. While not

Computers/Internet/Searching/Search_Engines	102548
Adult/Porn	99488
News	76599
Computers/Internet/On_the_Web/Online_Communities	57201
Shopping/Auctions	52293
Shopping	46558
Shopping/General_Merchandise/Major_Retailers	31559
Computers/Internet/E-mail/Web-Based	29369
Government	25456
Recreation/Travel	23098
Computers/Internet/Searching/Directories	23061
Reference/Education/Schools	22921
Reference/Encyclopedias	20408
Adult	19503
Business/Financial_Services/Banking_Services	17080
Arts/Movies/Databases	15712
Business/Real_Estate	14482
News/Newspapers	13890
Sports	13733
Arts/Music/Lyrics/Directories	11968

Figure 6.5: Top 20 activities performed by 4,188 AOL users.

definitive, this indicates a large portion of the population is focused on a relatively small area of interest. This chart further emphasizes the power of our approach to data normalization. Over 700,000 queries and click links were synopsisized into twenty fairly descriptive labels describing the activities of the users.

Unlike the browser history files, we found no national statistics for which we can compare our results. Figure 6.6 is a histogram showing the sessionization breakout for all 4,188 users. From this, we see the large proportion of the population (82%) had between 125 and 215 sessions. This is not overly surprising. Based on data from [136][118][23], we estimate the

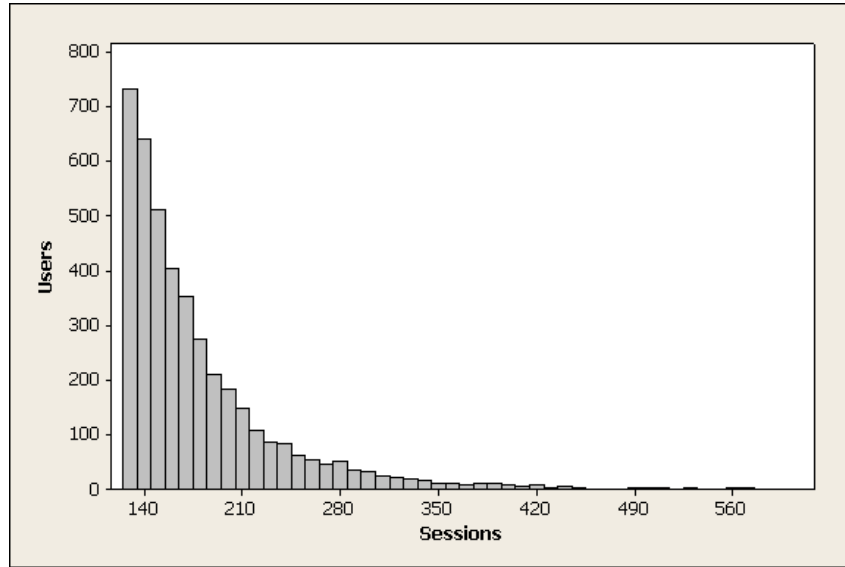


Figure 6.6: Sessionization histogram for 4,188 AOL users. Histogram emphasizes 82% of the population had 125-215 query sessions.

average number of browsing sessions per user per month to be approximately 30 and the average number of query based sessions per user per month to be 20 for 2006. Since we limited our user set to those having average or above query activity, these higher numbers seem reasonable.

6.2 Use Cases

The following three sub-sections detail use case scenarios in targeting, stress monitoring, and insider threat detection employed to test the validity of our behavioral modeling methodology. These scenarios were chosen due to their diverse domains as well as their visibility in the cyber domain. While extensive analysis was done for each use case, in an effort to minimize duplication of effort we only list results which make use of unique analysis techniques (i.e. if profiling is accomplished in scenario one and scenario two, we only detail the steps taken in one use case example).

6.2.1 Targeting

While the term *targeting* can be used in a number of contexts (i.e. targeting a consumer, targeting an insider, etc.), in this section we present an example of using our methodology to target from a more traditional, military definition of the term. The Department of Defense Joint Targeting publication 3-60 defines a target as “an entity or object considered for possible engagement or action. It may be an area, complex, installation, force, equipment, capability, function, individual, group, system, entity, or behavior identified for possible action to support the commanders objectives, guidance, and intent [59].” The goal of this section is to demonstrate our ability to instantiate and validate the profiling techniques outlined in Section 5.4.1 for the *individual*, *group*, and *behavior* portions of this definition. Mapping this to our system of systems behavioral modeling terminology as defined in Section 1.2, our goal here is to affect the user’s environment in some way and then determine if/how the user’s behavior changes (see Figure 6.7). Detection of this change, or lack thereof, is critical to determining both the effectiveness and level of effectiveness of the engagement.

Targeting involves selecting and prioritizing targets and matching the appropriate response to them, considering operational requirements and capabilities. Our emphasis of targeting is on identifying resources (targets) the opposition can least afford to lose or that provide the enemy with the greatest advantage, then further identifying the subset of those targets which must be acquired and attacked to achieve success. The joint targeting cycle (see Figure 6.8) is an iterative process (with some steps occurring concurrently) which offers insights regarding the steps that must be satisfied to successfully conduct targeting. While consisting of six phases, we are only interested in phases one and five. These two phases will be addressed in the following sections.

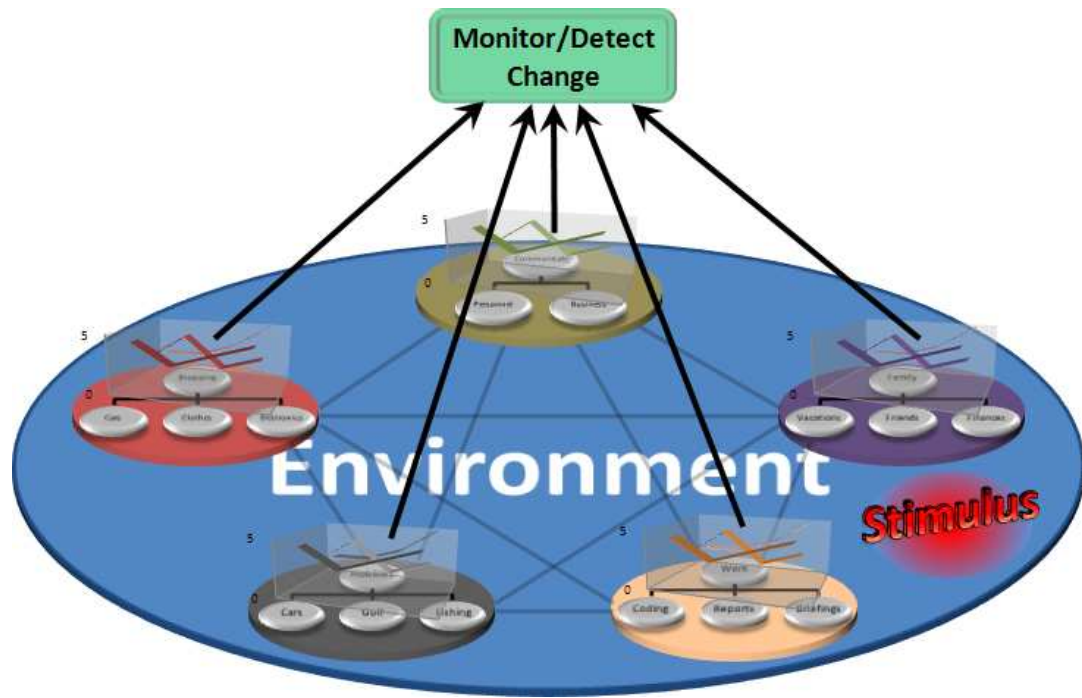


Figure 6.7: The goal of this scenario is to affect the environment in some way and then determine if/how the user's behavior changes.

6.2.1.1 Scenario

Phase one of the joint targeting cycle provides the initial impetus for the targeting process by outlining the specific end state sought via objectives and outcomes. The following will serve as phase one input.

The commander's objective in this scenario is to allow fighter aircraft to enter into enemy airspace and drop ordinance on previously selected military targets in the most effective and efficient manner possible. Key to this taking place is the disruption of the enemy's integrated air defense system (IADS). An IADS puts anti-aircraft sensors (e.g., radar, visual observers, and other technical means) as well as anti-aircraft weapons (e.g., anti-aircraft artillery, surface-to-air missiles, air superiority fighters and interceptors, etc.), under a common system of command, control, communications and intelligence (C3I). Disruption of this system is a critical aspect to the overall success of the mission.

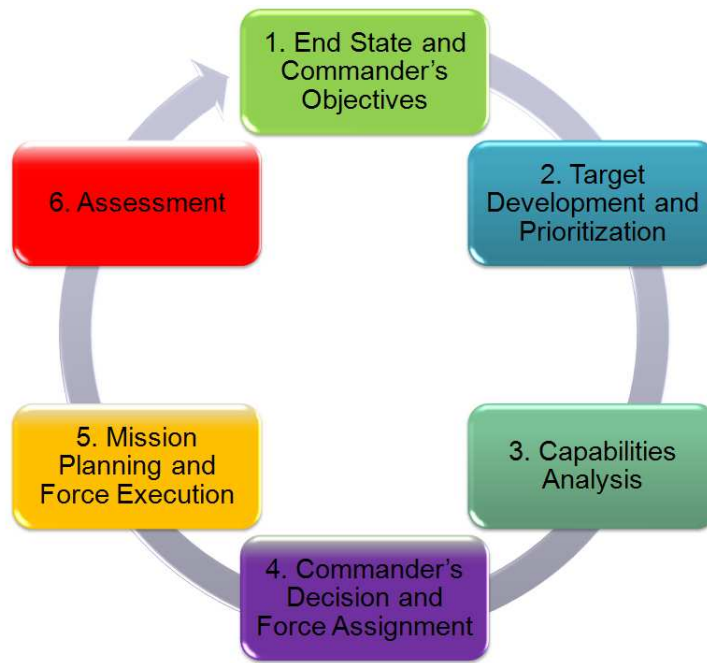


Figure 6.8: Department of Defense Joint Targeting Cycle

Intelligence reports show we have exploited a large number of computer systems on the air base responsible for the overall operation of the IADS, but currently have been unable to exploit any machines on the IADS itself. We also know the IADS overall command and control structure was designed and implemented by personnel working on the air base in question. With this in mind, the course of action (COA) is to identify the individuals on the air base responsible for coding the command and control system. Once discovered, it is believed access to these users' machines will afford greater insight into the command and control portion of the IADS thus providing options as to best deny, disrupt, or destroy the system using kinetic or non-kinetic means. Figure 6.9 is a graphical depiction of the key components of an IADS, highlighting in red the command, control, and communications component of interest.

With our objectives defined, we now move forward to phase five of the joint targeting cycle. Targeting in this phase consists of six steps: find, fix, track, target, engage, and assess

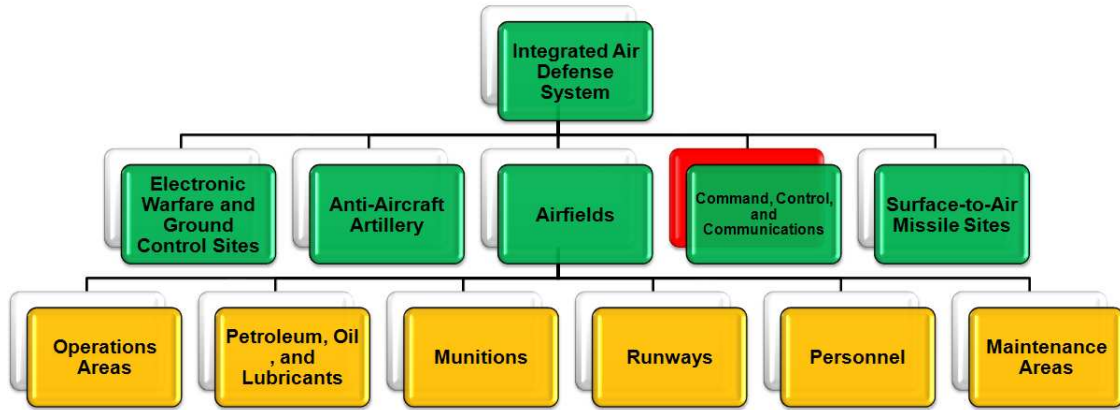


Figure 6.9: Graphical depiction of the key components of an Integrated Air Defense System (IADS). The command, control, and communications component (highlighted in red) represents the target of interest for this scenario.

(F2T2EA) (See Figure 6.10). The remainder of this section will outline how we make use of our methodology to achieve each of these steps. We will make use of the AOL dataset for this scenario build out and will assume the 4,188 users all work on the air base of interest.

6.2.1.2 Find

The goal of this step is to detect and classify targets for further prosecution. As previously defined, our targets of interest are computer programmers and our first step is to define a profile to identify them. We are not interested in those who may code as a hobby or to satisfy a school project, only those whose job it is to write code on a day to day basis. Initially, we are not interested in any specialty within programming (i.e. Java programmer, web programmer, etc.), but rather programmers in general.

Due to the complexity often associated with the task, programming is a fairly reference driven process often requiring extensive Internet searches. Because of this, programmers perform a large number of Internet searches, making the AOL data set a perfect match for this scenario. A benefit of targeting programmers is that they represent a fairly small percentage of the population. According to [155][48], only 0.0298% of all employed personnel

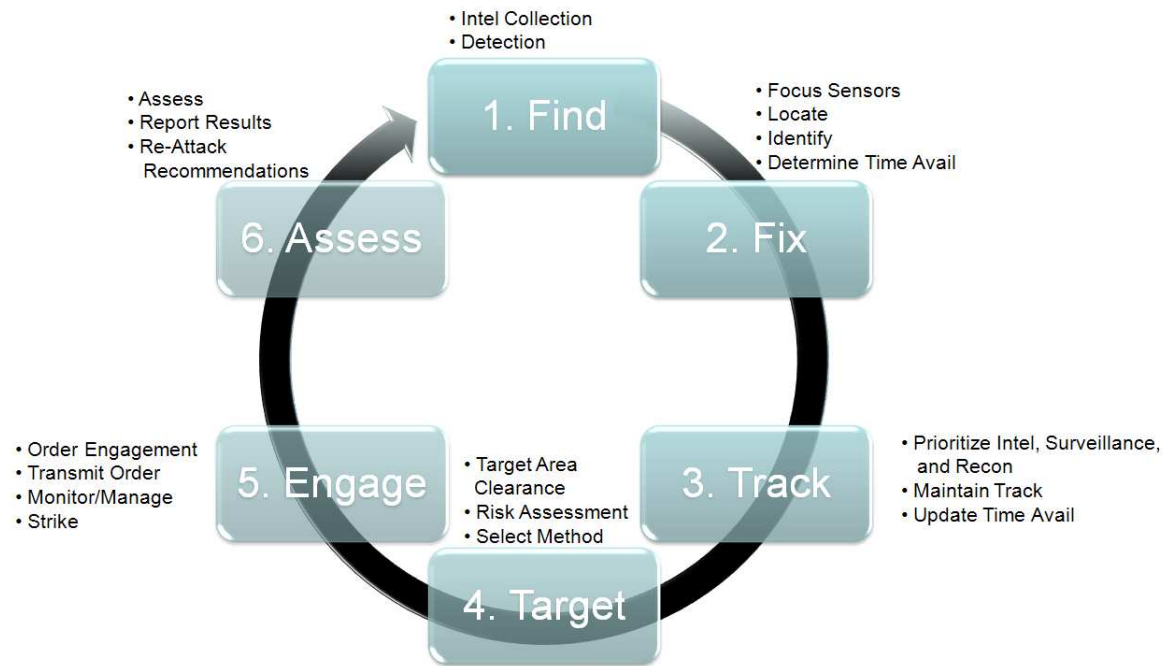


Figure 6.10: Phase five of the Joint Targeting Cycle; Find, Fix, Track, Target, Engage, and Assess.

in the United States were computer programmers working from home in the year 2006. Were we to target those users utilizing online communities (i.e. Facebook, MySpace, etc.) the result set could be very large and difficult to verify.

As discussed in Section 5.4.1, there are a number of characteristics we can choose from for our initial profile definition. We are looking for both a specific activity as well as persistence in that activity, so our profile attributes will be both contextual and temporal in nature. We use our six step process of Section 5.4.1.1 to create our profile.

Since “programming” is an obvious descriptive keyword for our target group, we query our ontology categories and category descriptions for references to the word. The top five categories returned are:

1. Computers/Programming
2. Computers/Artificial Intelligence/Genetic Programming

3. Computers/Parallel_Computing/Programming
4. Arts/Television/Schedule_and_Programming
5. Computers/Artificial_Intelligence/Programming_Languages

Other results returned are simply a subset of these. With the exception of item four, it is clear the combination of *Computers* and *Programming* seem to form the basis of what we are looking for and by ensuring both words are present, we eliminate false positive categories such as item four. Looking at “See also” categories, we also identify *Computers/Computer Science* as a category of interest. Using this information as our contextual basis, we identify 1,516 categories in our ontology consisting of some combination of *Computers* and *Programming* or *Computer_Science*.

To identify those who program for a living versus casual programmers, we add the profile constraint of only selecting individuals exhibiting profile persistence (as defined in Section 5.5.1) in forty percent or more of their sessions. While forty percent may seem somewhat low and arbitrary, as discussed in Section 5.4.1, profile definition can often be an iterative process. An alternate approach would be to identify all users with persistent profile browsing patterns and narrow our focus from there. We instead pre-filter any users with low persistent browsing behavior. To identify persistent profile usage within our data set, we do a per user linear least squares regression over all stable sessions based on the percentage of per session profile activity. Users are then rank ordered based on R^2 values.

The last aspect of the profile is also temporal and has to do with *when* individuals are performing profile activities. We make the assumption that someone coding for a living would be participating in this activity for extended periods of time (i.e. not just when they get home from work) and during daytime hours. While we have no empirical evidence to support this, we suspect a Gaussian distribution of profile related queries over a twenty-four hour period and add this as a stipulation to further focus our profile.

Upon instantiating our profile against the 4,188 users in our data subset, we identified 1,198 individuals having at least one query in the profile categories but only two having persistent profile interests. Both users having persistent profile interests had them for greater than forty percent of their sessions, thus meeting our profile criteria. We will refer to these individuals as “User 1” and “User 2” in order to maintain anonymity. We should note these two users represent 0.0477% of our sample population and differs from the previously stated national programmers statistic (0.0298%) by only 0.0172. While this similarity does not guarantee we have selected the correct users or we have not missed additional users, it does offer validation our profile results are reasonable. User 1 had 347 total sessions of which 171 contained the profile categories. Using our sample size estimation techniques outlined in Section 5.3, we determine that after session 102 we have collected enough data to characterize this individual. Figure 6.11 is a plot of our profile characteristics over stable sessions with a superimposed least squares regression plot (dashed red line). A R^2 value of 0.86, a Y

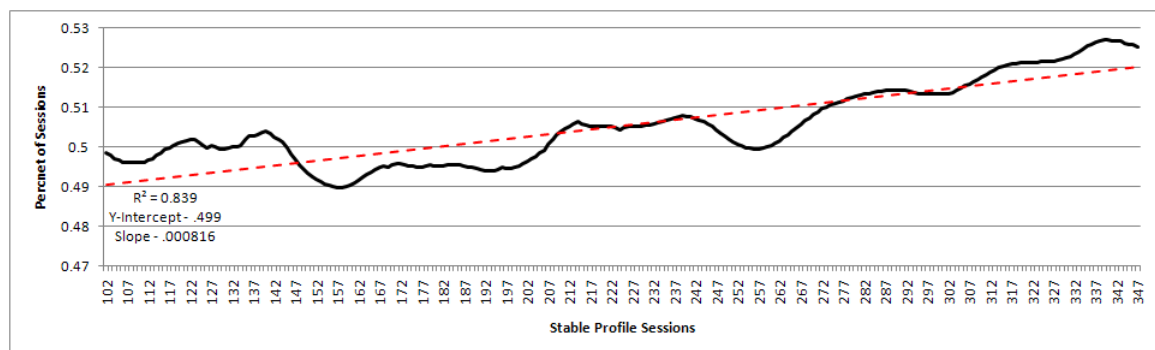


Figure 6.11: Plot of user 1 profile characteristics over stable sessions with a superimposed least squares regression plot (dashed red line).

intercept of 0.499 and a positive slope (0.70°) confirm our persistence requirements are met for this user. Although we identified two users meeting our profile criteria, we will only focus on User 1 for the remainder of this section and provide pertinent information regarding User 2 as appropriate.

6.2.1.3 Fix

A “fix” is a position determined from terrestrial, electronic, or astronomical data. In the cyber realm, this is typically an IP or MAC address. Inputs to this step of the targeting process are the potential targets (User 1 and User 2 from the previous step) while outputs are more detailed target identification, classification, and confirmation information. While we are confident in our choice of targets, detailed examination is needed to identify additional non-profile related information. This information may also provide additional profile characteristics which can in turn be used to expand or further refine our target pool. Figure 6.12 is a 1st order radial layout of the top level categories for User 1. The green root node

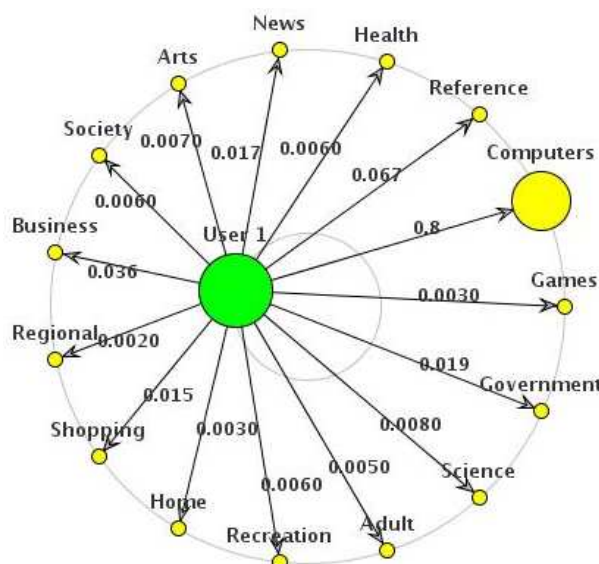


Figure 6.12: 1st order model of all top level activities for user 1. Size of activity nodes is based on the percentage of the root node represented (i.e. larger the node, larger the percentage)

represents the user while the yellow child nodes enumerate the individual root activity nodes. The size of the activity nodes is based on the percentage of the root node represented (i.e. larger the node, larger the representation of the root activity) while the edge numbers nu-

merically depict the proportion. As one would expect from our profile criteria, *Computers* represents the dominant activity (80% of all activities) of this user with *Reference* coming in a far distant second (6.7% of all activities). While a valuable first pass, this information provides no insight into the associated target activities of interest. We further break out *Computers* as shown in Figure 6.13 to examine finer grained activity details in this area. While fairly diversified, we see a large proportion (30%) of the user's *Computers* activity

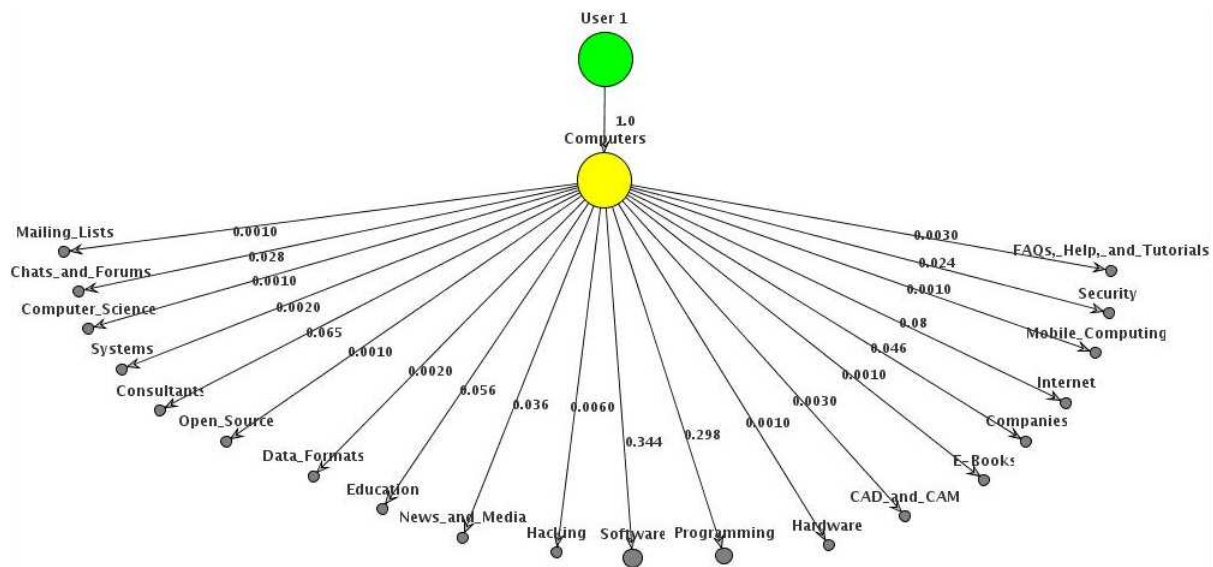


Figure 6.13: User 1 1st order model for the *Computer* activity.

is indeed focused in the area of *Programming* while the largest proportion is spent in the non-profile activity *Software*. To determine if *Software* contains information pertinent to our target profile, we expand our activity tree one additional level for the *Software* and *Programming* activities (Figure 6.14). It should be noted that the proportions of each node represented in Figure 6.14 are of the root activity and not of the parent node. For example, *Computers/Programming/Languages* represents 14.5% of all computer activities and *Computers/Programming* encompasses 46.4% of computer activities. Examination of the *Software* category provides informative data identifying specific areas of concentration

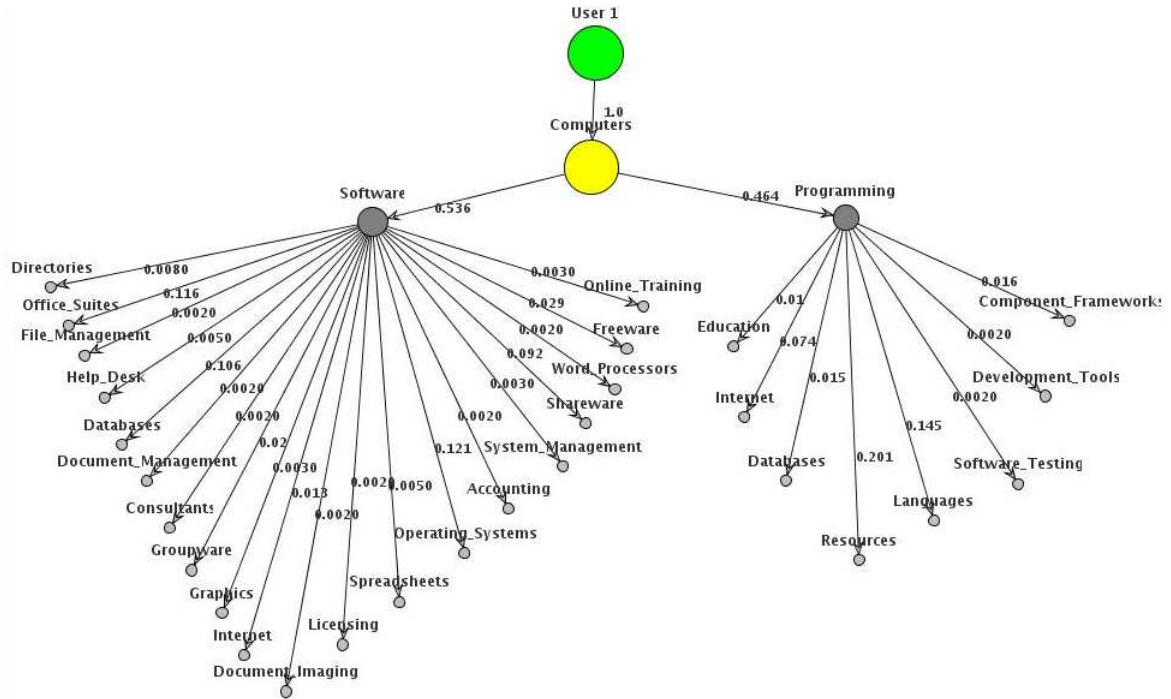


Figure 6.14: User 1 1st order model for of *Software* and *Programming* activities.

(deeper analysis shows 75% of these activities are focused on Microsoft operating systems, Microsoft Access, or Microsoft Office Suite), but does little to narrow our target profile in any way. If we later decide to only target users programming for Microsoft operating systems, this would be an obvious area to concentrate on.

While the 1st order models do an excellent job of outlining the proportion of time spent in individual activities, they do little to show the relationships between these activities. For this, we perform behavioral extraction as outlined in Section 5.1.1. Using these techniques we are able to identify and extract five predominant behaviors exhibited by this user. Figure 6.15 shows the distribution of activities associated with each identified behavior. Labeling is done manually and in a qualitative manner based on the general “theme” associated with the activities represented. We further differentiate the behaviors as being work related (shaded green) and non work related (shaded red) behaviors. While we have no ground truth information to verify this, based on the activities associated with each behavior, we

B1 - Education and Training	
reference/education	0.43838
computers/education/certification	0.33055
computers/programming/education	0.10316
reference/education/colleges_and_universities	0.02244
reference/education/schools	0.01786

B2 - Leisure	
computers/software/shareware	0.43411
computers/internet/searching	0.16279
computers/software/freeware	0.13953
computers/hacking/spyware	0.03101
adult	0.05263
computers/internet/chat	0.04211
games	0.04211

B3 - Programming	
computers/programming/resources	0.45423
computers/programming/languages	0.38526
computers/programming/component_frameworks	0.09205
computers/software/internet	0.02564
shopping/publications/books	0.02611
reference/books	0.01071

B4 - Intended Platform	
computers/software/operating_systems	0.26429
computers/software/office_suites	0.25357
computers/software/groupware	0.03133
computers/software/databases	0.16971
computers/programming/internet	0.16071
computers/chats_and_forums	0.09643

B5 - Troubleshooting	
computers/companies/product_support	0.41053
internet/on_the_web/weblogs	0.28421
computers/companies/ibm	0.15263
computers/software/help_desk	0.10158

Figure 6.15: Five behaviors associated with user 1 extracted using LDA. Behaviors shaded green represent work related behaviors while those colored red are non-work or leisure related behaviors.

are comfortable in making this delineation. Examination of the behavioral activities and their relationships provides no additional insights into our profile definition and further strengthens our analysis done to this point.

While we currently have no empirical mechanism to rate our profile effectiveness, the analysis just performed strongly supports our contextual profile selection criteria for identifying targets of interest in this scenario. Unfortunately we currently have no way to verify false negative results (computer programmers not identified by our profile criteria) and see this as an area of future research. As stated in Supplement One to the Commander’s Handbook for an Effects-Based Approach to Joint Operations, “an effects-based approach calls for a significant level of humility in expectations of certainty, precision, and control [217].” With this in mind, confidence is high the two users in question are programmers within the organization and are indeed confirmed targets.

6.2.1.4 Track

Once the targets are confirmed, the next step is to monitor activity and movements. There are two aspects associated with tracking in the cyber realm; tracking a computer and tracking a user. While sometimes the two are one in the same, there are instances where knowing

one does not guarantee knowing the other. If tracking a computer via network monitoring, dynamic Host Configuration Protocol (DHCP) may continually change the IP address associated with the computer of interest or the user may be using a laptop with the ability to connect to the network via a wired or wireless connection. Identification of the Media Access Control (MAC) address of the network adapter associated with the computer is helpful in sorting these issues out, but based on the type and location of network collection, this information may not be visible. If tracking the user, the individual may use a portable device capable of roaming, may use multiple computers within a given organization, or may use “hotspot” locations such as a library or Internet cafe. Using behavioral fingerprinting offers a mechanism to address these issues.

Using the query history of each targeted user, we can create a behavioral fingerprint of each individual using the techniques outlined in Section 5.4.2. Using these fingerprints, it is then possible to monitor multiple collection sources for user activity based on their fingerprint data. Because the data being used is dated and we are not actually tracking these individuals, we will not go through the process of creating behavioral fingerprints, but simply note the huge potential offered by this technique in this scenario. The ability to identify when and where a person is online based solely on their cyber behaviors is an area of critical importance in cyber targeting. Using this approach provides a mechanism to determine when and where a user is with probabilistic certainty.

In addition to tracking a user’s physical location, the ability to track what a user is doing in the cyber realm and when it is being done is of significant importance to this particular course of action. As our focus is in the user’s programming habits, we only track work related behaviors (behaviors B3, B4, and B5 of Figure 6.15). Figure 6.16 shows the proportion of time these behaviors are exhibited over all stable sessions. From session 102 to 348, the state of the user is determined to be in PBS $P1$ (slope in degrees is -0.69°) and is depicted on the figure as a dotted black line. Upper and lower confidence bounds are calculated and

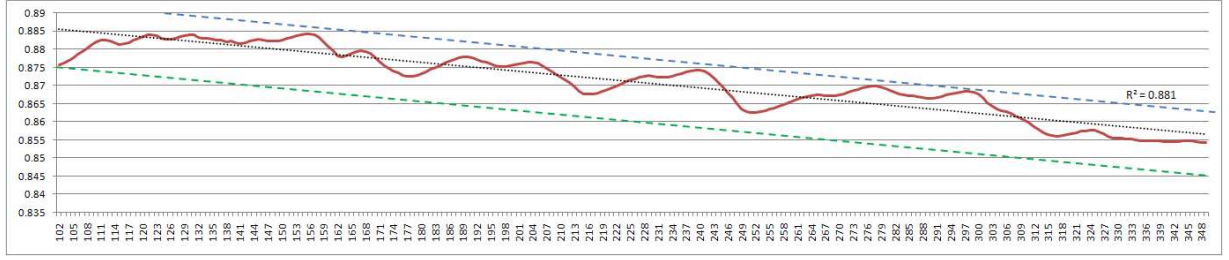


Figure 6.16: Plot of the proportion of time user work behaviors are exhibited over sessions for User 1 (red line) with a superimposed least squares fit (dashed black line). Upper and lower confidence bounds depicted as dashed blue and green lines respectively.

added to the plot as dashed blue and green lines respectively. Given the current state of the user, using techniques outlined in Section 5.5.1.1, predicting future states is accomplished by merely extending our linear fit from the last observation forward using the same calculated slope value.

6.2.1.5 Target

Once the target is identified, classified, and located a decision as to when and where to engage must be made. Depending on the type of cyber effect to be achieved, it may be beneficial for the user to be active on the computer or it may be just as important the user not be active on their machine. Determining this time frame is critical to achieving the objective and often very difficult to assess. As stated in our original profile definition, one of our requirements/assumptions was the target users have a Gaussian distribution relating to their computer usage. To verify this, we mapped the timestamps of the query sessions along with individual queries themselves onto a histogram. These graphs allow us to confirm the Gaussian nature of our users activities and provide key insights regarding the optimal time frames to engage the individual targets. Figure 6.17 is a 1st order histogram of the proportion of User 1's individual queries mapped onto a twenty four hour scale. We used the Anderson-Darling and Kolmogorov-Smirnov tests to verify the normalcy of the distribution.

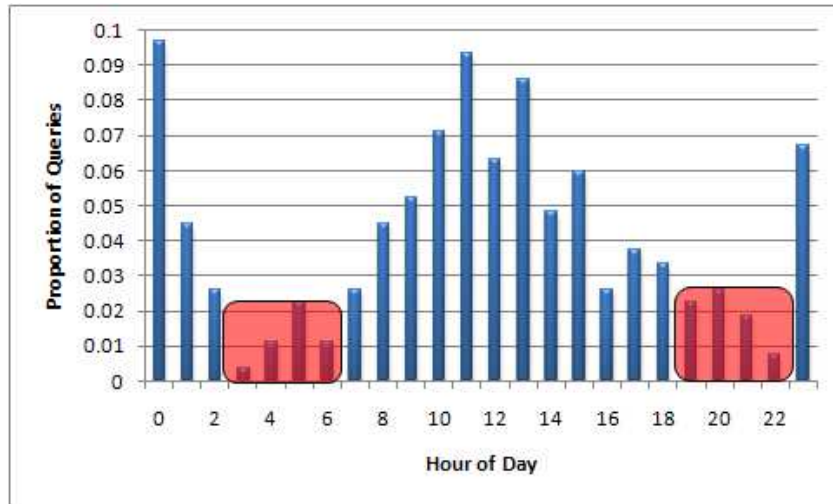


Figure 6.17: 1st order histogram of the proportion of user 1's queries made over a twenty four hour period.

Of interest are the significant “down” periods from 0300 till 0700 and 1900 till 2300. Perhaps even more interesting however, is the period of activity from 2300 to 0200. For most individuals this would be considered a time to sleep and may have been considered as a viable attack window in certain covert attack scenarios. As shown in Figure 6.17, this is a consistently active period of time for this individual and would be a poor time to plan an effects based operations reliant on no one being online. Temporal activity at this time may also imply remote connectivity. If the collection point for our data is at the individuals place of work, one would not expect late night access unless it was being done via remote connectivity. This information may also be of significant interest depending on the type of attack being performed.

With a confirmed target set as well as defined windows of opportunity, the next step in the process is to *Engage*. This step is primarily concerned with the actual attack itself against the target and while some opportunities exist to use our methodology here, in general this is an area primarily focused on the actual strike itself. Because of this, we will skip this step of the process and move on to the assessment step.

6.2.1.6 Assess

During the assessment phase, an evaluation of the results of the engagement is conducted to determine whether the desired outcome was achieved. Ideally the assessment should provide quick results allowing for expeditious re-attack recommendations. Unlike in the kinetic realm where assessment involves confirmation of physical destruction, assessment in the cyber realm is more subtle. Our overall goal is to identify desired and/or undesired changes in a user's behavior based on targeting activities performed. One mechanism to do this is to monitor deviations from our baseline fingerprint as outlined in Section 5.6.3. While Kullback-Leibler is one mechanism to measure these deviations, for this experiment we track the cumulative difference between each category proportion over time to its initial fingerprint proportion. Figure 6.18 is the plot of this difference over sessions. While the plot depicts an apparent

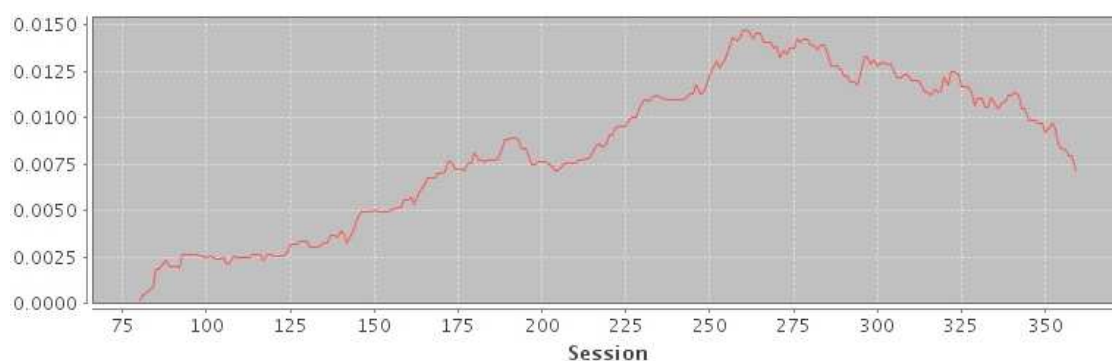


Figure 6.18: Plot of cumulative differences from baseline fingerprint proportions for each category for User 1. Relative lack of deviation indicates the individuals behavior did not change significantly during this time period.

significant change in overall behavior from session seventy five to session two hundred and sixty, the scale of the change is insignificant and does not even show up on our CUSUM charts. If the goal of the cyber effect was to not be detected by the target user, this chart would be one indicator of a successful mission emphasized by the lack of change in the user's overall behavior.

6.2.2 Cyber Stress Indicators

Next we will demonstrate how our methodology is employed for stress detection. The goal of this experiment is to determine if *online data* can identify and model a human characteristic not normally associated with computer interaction; stress. Where the goal of the previous scenario was to monitor for change based on a known stimulation to the environment, with these experiments we are looking to detect and identify the stimulation from detected behavioral changes (Figure 6.19). This technique is applied to both browser history files and

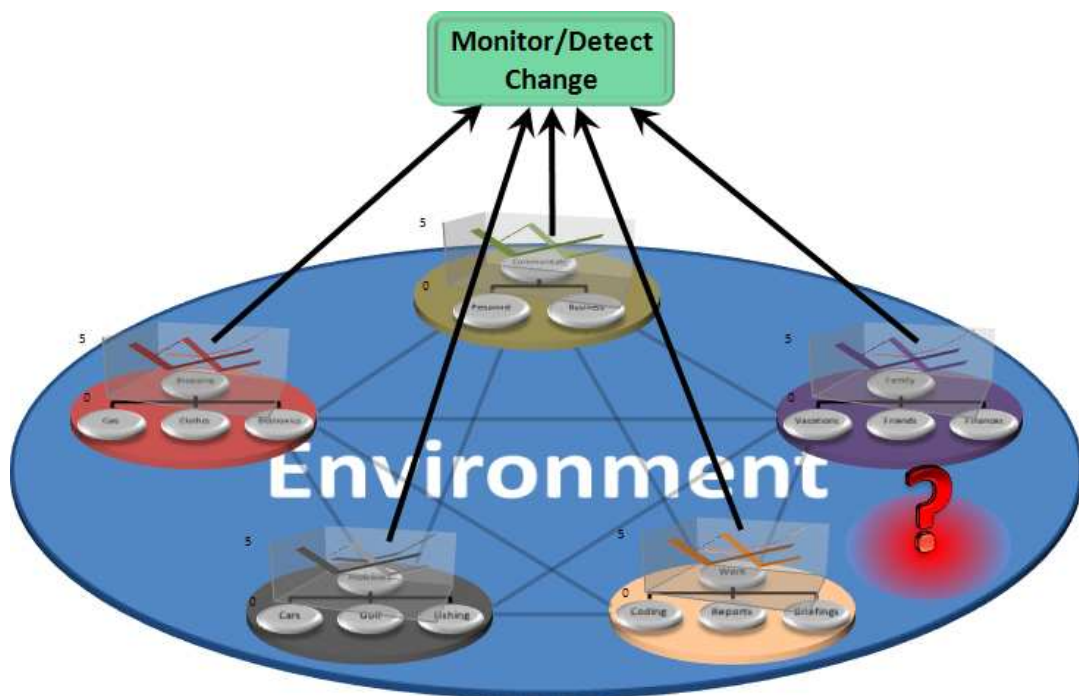


Figure 6.19: The goal of cyber stress monitoring is to detect and identify stimulation to the environment from behavioral changes to individuals and groups.

the AOL data set as defined at the beginning of this chapter.

While it seems there is very little consensus on a formal definition, stress is normally categorized in two way; negative stress and positive stress. Negative stress is a condition in which extreme pressure, hardship, or pain is either suddenly experienced or builds up

over time. This type of stress can range from anxiety or concern to mental or physical problems. Positive stress, on the other hand, can affect a person by stimulating awareness and motivation and can lead to increased motivation, more focused energy and improved performance. These top level categories are further broken down into the following four types of stress [199]:

1. Eustress
2. Distress
3. Hyper-stress
4. Hypo-stress

Eustress is a positive form of stress, usually related to desirable events in a person's life. It is this type of "positive stress" which provides us a feeling of fulfillment or contentment and also makes one excited about life. Below are some examples of eustress.

- Having a child
- Receiving a promotion or raise at work
- Taking a vacation
- Planning a wedding

In contrast, Distress, or "negative stress", is characterized by feelings of unfamiliarity and discomfort. Distress is caused by adverse events and relates to a person's inability to cope in certain situations. Some events leading to distress are:

- Death of a loved one
- Financial problems
- Strained relationship
- Chronic illnesses

Hyperstress is also normally seen as a “negative stress”, caused when a person is pushed beyond what he or she can handle. Hyperstress results from being overloaded or overworked. While this type of stress can be positive (i.e. adrenaline rush to finish a big project), long term exposure in such an environment can be very harmful. Hyperstress is often caused by the following:

- Excessive job demands
- Unproductive and time-consuming meetings
- Multi-tasking
- Working in fast paced environment
- Writing a Ph.D. thesis

Hypostress is the direct opposite of hyperstress and is experienced by a person who is under stimulated and constantly bored. Someone in an unchallenging job (i.e. factory worker performing the same task over and over) will often experience hypostress. The effect of hypostress is feelings of restlessness and a lack of inspiration.

This work will focus on distress and hyperstress since, as we shall show, contain characteristics which manifest in uniquely classifiable cyber behaviors.

6.2.2.1 Distress

Distress was chosen because of its visibility in the cyber spectrum. Stress relief is something which must be searched for. A user experiencing distress will look to research the cause and find ways to minimize the effect. The World Wide Web (WWW) offers a direct and easily accessible source of information to users and can be queried using any number of search engines. These queries and the associated web sites visited contain a great deal of information pertaining to the stressful event. We look to utilize this information as *stress indicators* to

determine both when and what type of distress is being experienced. While these indicators do not guarantee a user is experiencing stress, identification and characterization of this data, combined with other external factors (i.e. constantly late to work, productivity drop off, etc.), provides a mechanism to model this type of behavior in the cyber realm.

To identify these cyber *stress indicators* we rely on the Holmes-Rahe Life Stress Scale. In 1967, psychiatrists Thomas Holmes and Richard Rahe examined the medical records of over 5,000 medical patients to examine correlations between stressful events and illnesses. Patients were asked to rate forty-three life events utilizing a stress scale. The results of the data collected showed a positive 0.1 correlation between their life events and their illnesses. These results were later published as the Social Readjustment Rating Scale (SRRS)[87], but are more commonly known as the Holmes and Rahe Stress Scale. Figure 6.20 is a sample of the scale showing the stress “points” associated with each stressor. To measure stress, a

Life Event	Mean Value
1. Death of spouse	100
2. Divorce	73
3. Marital Separation from mate	65
4. Detention in jail or other institution	63
5. Death of a close family member	63
6. Major personal injury or illness	53
7. Marriage	50
8. Being fired at work	47
9. Marital reconciliation with mate	45
10. Retirement from work	45
11. Major change in the health or behavior of a family member	44
12. Pregnancy	40
13. Sexual Difficulties	39
14. Gaining a new family member (i.e.. birth, adoption, older adult moving in, etc)	39
15. Major business readjustment	39
16. Major change in financial state (i.e.. a lot worse or better off than usual)	38
17. Death of a close friend	37
18. Changing to a different line of work	36
19. Major change in the number of arguments w/spouse (i.e.. either a lot more or a lot less than usual regarding child rearing, personal habits, etc.)	35
20. Taking on a mortgage (for home, business, etc..)	31

Figure 6.20: Top 20 Holmes-Rahe Stressors

tally of points related to specific stressful *Life Events* occurring in the past year are added together, determining the likeliness of contracting a serious illness in the immediate future.

We use the same *Life Events* listed on the scale as a basis to identify stress-related categories within our training ontology. By doing keyword searches on these events, as well as through manual examination of the data set, we were able to identify forty representative stress categories (see Figure 6.21). Of note are the four categories ending with a percent (%)

Business/Financial_Services/Loans	Shopping/Weddings
Business/Financial_Services/Mortgages	Shopping/Death_Care
Business/Investing/Retirement_Planning	Society/Crime/Abuse
Health/Addictions	Society/Crime/Sex_Offenses
Health/Mental%	Society/Death%
Health/Conditions%	Society/Issues/Children_Youth_and_Family/Adoption
Health/Reproductive_Health/Clinics_and_Services	
Health/Reproductive_Health/Pregnancy_and_Birth	Society/Issues/Children_Youth_and_Family/Child_Abuse
Health/Support_Groups	Society/Issues/Violence_and_Abuse
Health/Teen_Health/Drugs_and_Alcohol	Society/Law/Legal_Info/Family_Law/Divorce
Health/Teen_Health/Teen_Pregnancy	Society/Law/Legal_Information/Bankruptcy
Home/Family/Pregnancy	Society/Law/Legal_Information/Drunk_Driving
Home/Family/Adoption	Society/Law/Products/Bankruptcy
Home/Personal_Finance/Money_Management/Debt%	Society/Law/Products/Self-Help/Bankruptcy
Home/Personal_Finance/Retirement	Society/Law/Products/Self-Help/Family_Law/Divorce
Kids_and_Teens/Teen_Life/Suicide	Society/Law/Services
Kids_and_Teens/Your_Family/Adoption	Society/People/Men/Widowers
Shopping/Health/Conditions_and_Diseases	Society/People/Women/Widows
Shopping/Health/Mental_Health	Society/Relationships/Anger_Management
Shopping/Health/Reproduction_and_Sexuality	Society/Relationships/Divorce
Shopping/Health/Substance_Abuse	Society/Relationships/Weddings

Figure 6.21: Forty stress categories from our activity ontology with a direct relation to Figure 6.20.

sign. This is used as a wild card of sorts to represent the category and all sub-categories below it. For example, *Society/Death%* includes *Society/Death*, *Society/Death/Death_Care*, *Society/Death/Issues*, *Society/Death/Suicide*, and a number of other sub-categories, bringing the total number of categories used to ninety-three. With this categorical mapping in hand, we now have the means to relate distress indicators to cyber activities.

Our goal in this section is to identify a distress related event for a group or individual within the data set. Group distress may be triggered by widespread news of a disease outbreak, terrorist attack, large layoffs within an organization, or an incredibly poor day on Wall Street. Although we have no empirical evidence to prove this, we believe group distress

will normally be characterized as being transient in nature (as defined in Section 5.5.1). In this situation, we would expect above average cyber activity in a distress related area by a significant number of individuals, but we do not see this distress commonality being exhibited for any significant period. One area we see this type of group analysis proving useful is in monitoring the health and wellness of a given population. Individual distress can be chronic or acute in nature and will be characterized as such by persistent or transient query activity in one or more distress related categories. Analysis of individual stress could provide a mechanism of monitoring individual health in certain circumstances. The Army is currently facing significant increases in suicides within its forces [236]. To combat this, they could use this technique to monitor soldiers' Internet usage in stress related categories to help identify those exhibiting distress and intervene as necessary.

As stated previously, we will use the AOL data set for this experiment. The first level of analysis performed involves identifying counts for stress-related activities within the data. This subset of the original data consisted of 2,517 users (60.1% of the original population) who executed 18,728 stress related queries (1% of the total searches). It should be noted, we are not implying by these numbers that 60.1% of the population is exhibiting distress. The histogram shown in Figure 6.22 emphasizes this fact and shows 1274 (51%) of the users executed four or less stress related queries and 2,133 (85%) had twelve or fewer stress-related queries over the three-month period. Figure 6.23 is a break out of the top twenty stress labels queried within the population and their associated counts. In a real world implementation, one would assume some level of insight into the population and environment being monitored and categories of interest would be identified for which they would wish to monitor as well as context regarding events happenings during the timeframe of interest. Without background on our population or their surroundings, we decided to analyze each of the top twenty categories.

We have defined a number of techniques capable of identifying both distressed individuals

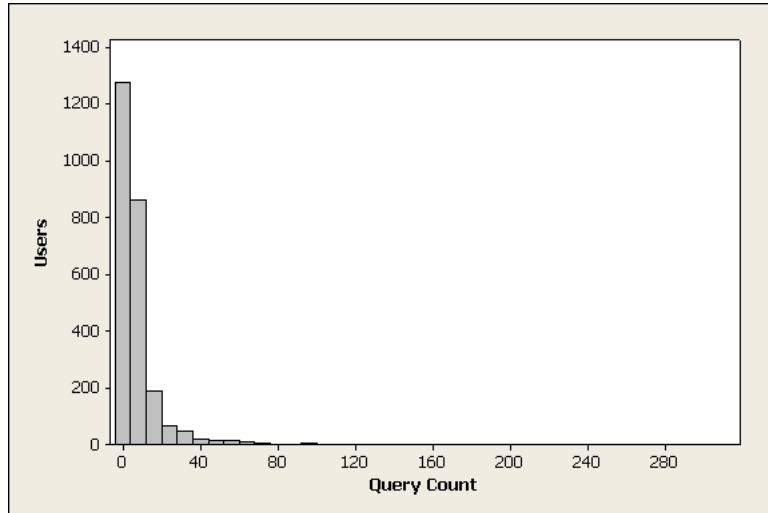


Figure 6.22: Histogram showing the number of users associated counts of stress related queries.

or groups of interest within the data. Clustering could be used to group users with similar distress characteristics or profiles created to select only those users having distinct distress traits. As stated earlier, the analysis performed greatly depends on the desired goal or outcome. With this in mind, we use statistical process control based anomaly detection techniques (see Section 5.6.1) to identify categories exhibiting abnormal characteristics. To filter the list to those categories which appear “interesting”, we generated a queries per day CUSUM of each stress category listed in Figure 6.23 and rank order the results based on the number and significance of the deviation. Our original goal in taking this approach was to identify group level stress within the population based on abnormally high query counts. *Health/Mental_Health/Disorders/Anxiety* was ironically the highest ranking item on the list by a significant margin. Figure 6.24 is the CUSUM representation of this data. Days one through fifty six all exhibit relatively stable query counts followed by significant spikes on days fifty nine and sixty six. At first glance, it would be easy to make the assumption the observed spikes in interest were transient in nature and possibly related to a group stressor affecting users in the data set. Figure 6.25 shows the top ten breakout of counts within this

Health/Conditions and Diseases/Cancer	1472
Shopping/Weddings	1299
Health/Conditions and Diseases	1250
Health/Mental Health/Disorders/Anxiety	1046
Home/Family/Pregnancy	769
Health/Conditions and Diseases/Musculoskeletal Disorders/Back and Spine	664
Health/Reproductive Health/Pregnancy and Birth	571
Health/Mental Health/Disorders/Resources	549
Business/Financial Services/Loans	405
Health/Conditions and Diseases/Immune Disorders/Immune Deficiency/AIDS	212
Health/Mental Health/Policy and Advocacy	198
Society/Death/Death Care/Cemeteries/Locating Graves	198
Health/Mental Health/Disorders/Directories	197
Health/Mental Health/Counseling Services/Online/Advice	188
Health/Conditions and Diseases/Cancer/Breast	183
Health/Conditions and Diseases/Endocrine Disorders/Pancreas/Diabetes/Organizations	183
Home/Family/Adoption	176
Health/Conditions and Diseases/Infectious Diseases	171
Health/Addictions	167
Health/Mental Health/Disorders/Mood/Bipolar Disorder/Treatment	165

Figure 6.23: Listing of the top twenty stress related categories (and associated counts) within the AOL data set.

category per user. While there were 192 users who made queries in this category during the three month period, 84% of the searches were in fact made by one individual. Although this was not the group distress event we were originally searching for, this approach did identify an individual of significant interest. To correct for this from happening in the future, we use a variance measure on the user counts to identify evenly dispersed distress events. Using our sample size estimation techniques outlined in Section 5.3, we determine that after session 123 we have collected enough data to have a 95% confidence of being within 10% of the mean for each category for this individual.

While the activity description is fairly detailed (*Health/Mental Health/Disorders/Anxiety*), it still gives little indication of the environmental factor or the *what* associated with this stressor. By extracting the query terms associated with this user from our behavioral activity database, we obtain the below results. While Figure 6.26 only shows the top ten query terms, in total, 872 of the 1871 (47%) queries by this user were related to selective mutism. Selective mutism is indeed a severe anxiety disorder (as indicated by our activity label) normally found in children in which the individual, who is normally capable of speech,

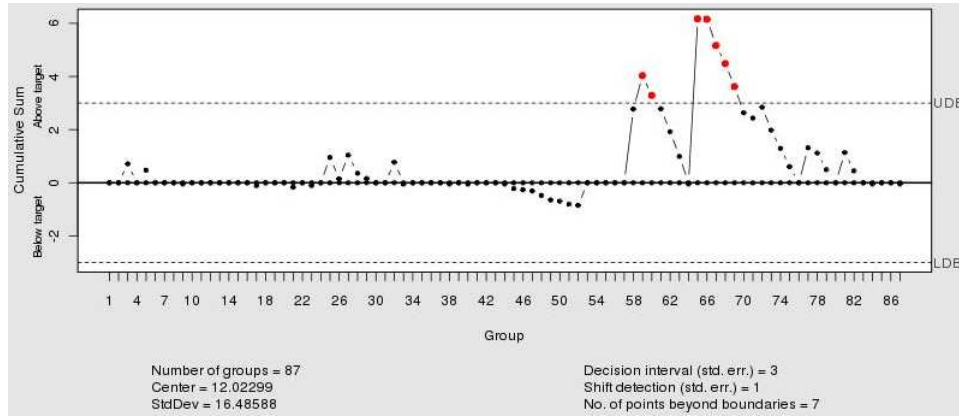


Figure 6.24: CUSUM chart of the number of *Anxiety* related queries over time. Anomalous counts are shown on days fifty nine and sixty six.

User	Count
1	878
2	17
3	16
4	13
5	12
6	9
7	8
8	8
9	7
10	7

Figure 6.25: Top ten listing of the number of *Anxiety* related queries made per user.

is unable to speak in given situations or to specific people. While 47% of the queries and over 60% of the sessions are dedicated to this disorder, we still have no insight into the behaviors exhibited by the individual making the queries. Using our behavioral extraction techniques of Section 5.1.1, we identify three primary distress related behaviors associated with this user. Figure 6.27 is a breakout of the distribution of activities associated with each identified behavior. These three behaviors represent 72% of all queries made by this user. Behavior one is entirely focused on the distress activity itself; in this case selective mutism. We see this as a kind of “blind search” behavior for anything and everything associated with selective mutism. Behavior two is much more diversified than the other two and seems to address coping with the distress for both the individual and the family members. The

Query	Count
selective mutism	545
selective mutism and learning problems	50
selective mutism and gifted	33
selective mutism and gifted children	23
effects of selective mutism on learning	22
selective mutism or neurological delay	22
selective mutism and learning disabilities	21
selective mutism and language problems	20
selective mutism and academic problems	20
selective mutism and learning gaps	18

Figure 6.26: Top 10 anxiety related query terms for user 1 of Figure 6.25

Behavior 1	
health/mental_health/disorders	0.99149

Behavior 2	
health/child_health/growth_and_development	0.34904
health/medicine	0.14122
internet/on_the_web/weblogs	0.12599
health/child_health/information_and_advice	0.10061
home/family/parenting	0.07061
health	0.03553
reference/education/products_and_services	0.03553
health/resources/for_patients	0.0203
health/medicine/education	0.0203
health/conditions_and_diseases	0.01523

Behavior 3	
reference/education/special_education	0.82342
health/professions/speech_therapy	0.08266
health/medicine/facilities	0.03266
health/mental_health/child_and_adolescent	0.01899
health/pharmacy/pharmacies	0.01266
health/mental_health/organizations	0.01266
health/child_health/information_and_advice	0.00633

Figure 6.27: Breakout of the distribution of activities associated with each behavior associated with our “selective mutism” user.

activities associated with this behavior seem to focus on finding a way for both the family and child to deal with the distress. Behavior three is centered primarily on education and speech therapy as a means to alleviate the distress and is highlighted by the concentration in the *Special Education* and *Speech Therapy* activities. While examination of queries only would have identified selective mutism as the dominant theme for this user, it is not until we decompose queries into activities into behaviors that we see three very distinct aspects of this one focus area. Figure 6.28 is a stacked graph of the three behaviors over time. The y

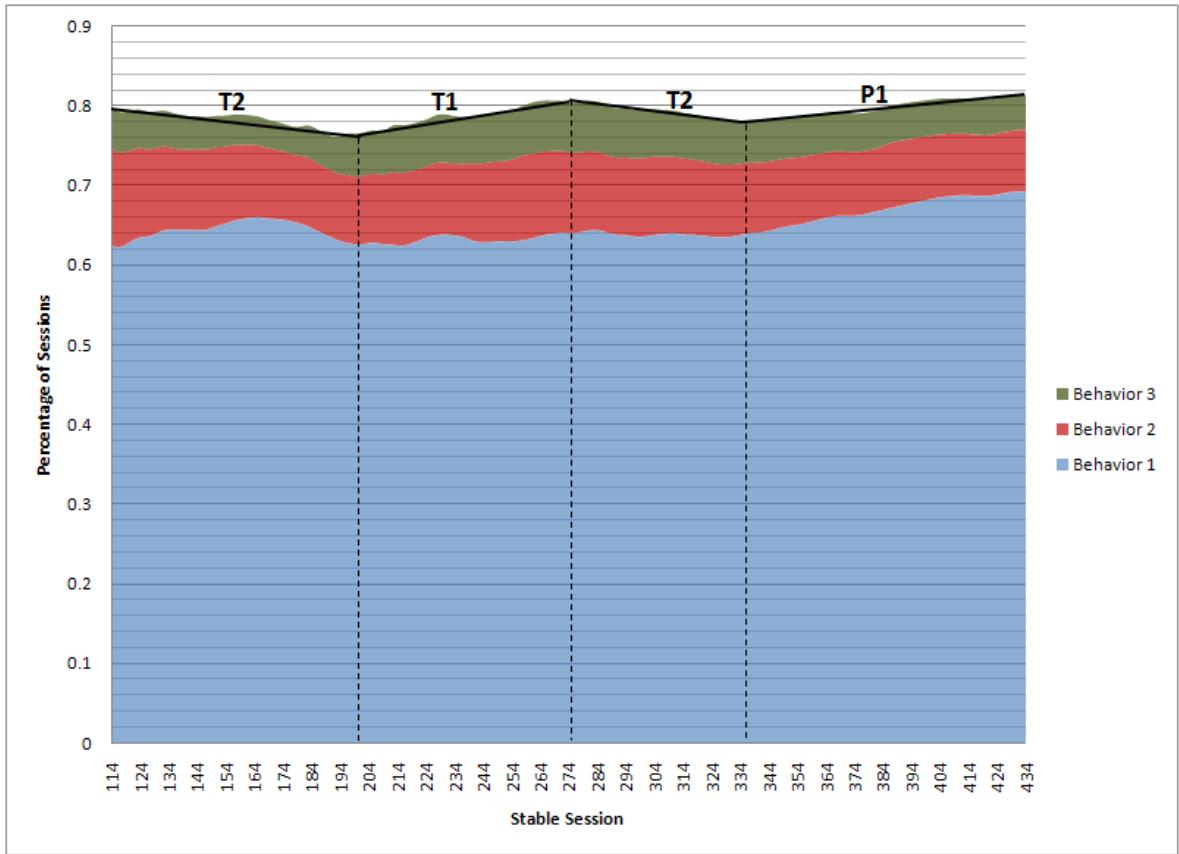


Figure 6.28: Stacked graph of distress related behaviors over time.

axis represents the percentage of sessions focused in each behavior, the dashed lines highlight the break points determined by doing piecewise linear approximation (see Section 5.5.1.1), and the straight lines framing behavior three depict the result of the linear approximation. The text above each linear segment represents the state of the segment where P represents a persistent state and T a transient state. As indicated by the x axis values, 113 sessions occurred before all activities associated with these three behaviors stabilized within 10% of their true mean. While not highlighted on the graph, by session 314, our adaptive filtering mechanism had readjusted the window size from the original value of 113 to 99 thus allowing the fourth behavioral state to be labeled as persistent. Figure 6.28 represents a graphical synopsis of our ability to identify and track stressful behaviors over time. Additional anal-

ysis in this area could include temporal analysis (time of day they make queries) or profile definition and clustering (to find additional users with this “type” of distress). We will not cover these items as they are addressed in other scenarios within this chapter.

6.2.2.2 Hyperstress

In order to model hyperstress, we take advantage of the “running around like a chicken with its head cutoff” phenomenon often exhibited by individuals in this situation. This is often characterized by users performing activities at a pace well above average for an individual. While this situation is difficult to quantify and qualify in the physical realm, it is easily captured and monitored in the cyber arena. For those users who rely on computers for the preponderance of their job requirements, this abnormal activity presents as a dramatic increase in application usage or web browsing. Someone working to meet a deadline is observed suddenly increasing their usage on a particular application associated with building a presentation or writing a report. This same suspense could also present itself through an upsurge in web activity needed for online research to finish the briefing or report. Just as increases in application or network traffic are indicators of hyperstress, so are dramatic decreases in these areas. A user who normally surfs the web first thing in the morning and during breaks throughout the day might suddenly stop this activity altogether in order to complete a time sensitive tasking. It is these dramatic increases and decreases which we will leverage to determine when hyperstressful situations are taking place.

There are two situations in which cyber analysis depicting changes in routine are most effective. We could monitor all traffic for a user or group of users and look for significant increases or decreases in volume (i.e. total click-links, queries, links per session, etc.). If a large increase in volume is detected, LDA techniques (Section 5.1.1) can be employed to determine the specific categories/behavior related to this surge. If a large decrease in volume is detected, one would want to determine the categories of information missing in

order to help rationalize the drop (i.e. sudden drop in all programming related traffic may mean coders are at an offsite that day). A second situation in which this technique is useful involves targeted profiling. Instead of monitoring all traffic for all users, profiles would be defined to identify specific users or groups of interest and traffic volume for these individuals or groups could then be monitored for patterns of interest. As an example of this, using our programmer profile from Section 6.2.1, we could monitor traffic for just the programmers within the air base to identify significant increases/decreases in programming related queries. In this case, significant increases in activity might signify intensified workload related to new projects or software upgrades.

Individual Hyperstress

For this section, we analyze subject's browsing history data to identify a specific instance of individual hyperstress. Of the users for whom we have data, one identified a particularly stressful situation occurring during the time period of data capture; defense of thesis topic proposal. Our goal here is the quantitative identification of the "when" and "what" associated with the event.

Our user (whom we will refer to as "User 1" for the remainder of this section) is a graduate student conducting research in the areas of finance and computer engineering. Browsing history data for this individual was collected from 28 October 2009 to 7 January 2010 and consists of 10,907 click links, 623 sessions, and sixteen top level categories. As we are interested in hyperstress identification, we begin by examining the user's temporal browsing activities. For this, we examine the temporal aspects of how often the individual is using the Internet. Figure 6.29 is a plot of the number of URLs visited per day for the user. While anomalous activity is observed around days thirty and fifty five, the source of these spikes and their relation to our subject's hyperstress is indeterminable. Using behavioral extraction (see Section 5.1.1), we are able to identify and label seven dominant behaviors for this user. Figure 6.30 is a breakout of the distribution of activities associated with each

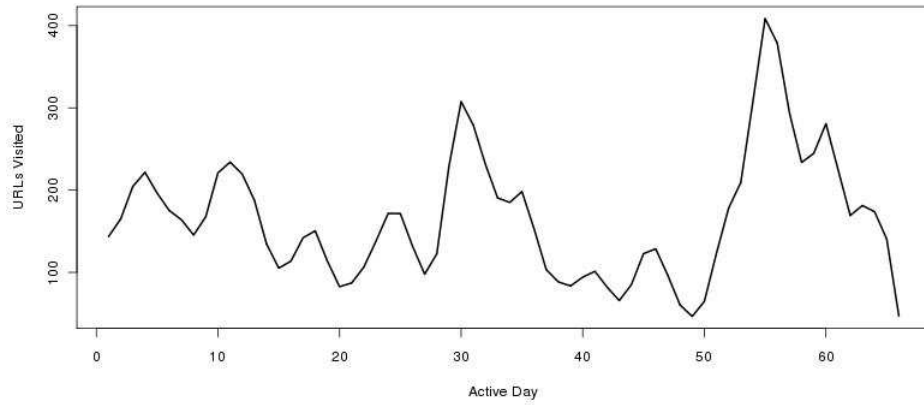


Figure 6.29: Two month plot of the number of URLs visited per day for our hyperstress candidate.

identified behavior. The labels are defined in a qualitative manner by simply choosing a

B1 - Shopping		B2 - Online Communities		B4 - Email & General Interests	
shopping	0.38802	computers/internet/on the web	0.98929	computers/internet/e-mail	0.59913
shopping/auctions	0.1849			news	0.20408
shopping/sports	0.11458			sports/resources/news_and_media	0.04446
shopping/recreation/sporting_goods	0.05729			sports/skiing/killington	0.02114
recreation/travel	0.03906			reference/education/colleges_and_universities	0.01749
shopping/general_merchandise/major_retailers	0.03125			arts/music/bands_and_artists	0.01458
business/telecommunications/carriers	0.02604	B3 - Job Hunting		sports	0.01166
computers/software/freeware	0.02344	business/employment	0.53695	reference/maps/google_maps	0.00802
shopping/recreation/outdoors	0.02344	business/employment/resumes_and_portfolios	0.23167		
		business/employment/careers	0.18227		
B5 - Investing		B6 - Programming		B7 - Research	
business/investing/brokerages	0.25482	science/math/software	0.32292	computers/internet/search_engines/academic	0.56562
computers/software/databases	0.20343	computers/programming/languages	0.29688	reference/knowledge_management/publications	0.20938
business/investing	0.18951	computers/software	0.15104	science/publications	0.06406
business/investing/stocks_and_bonds	0.13704	computers/software/operating_systems	0.13542	science/astronomy/publications	0.03438
business/investing/day_trading	0.06103	computers/chats_and_forums	0.09375	science/publications/journals	0.02969
business/investing/associations	0.04283			computers/e-books	0.02344
business/investing/software	0.02463			science/math/publications	0.01875
business/financial_services/investment_services	0.02034			reference/libraries/library_and_information_science	0.01875
business/financial_services	0.01392				
business/investing/news_and_media	0.01392				

Figure 6.30: User 1 personal (blue) and work (green) behaviors extracted using LDA.

central theme based on manual browsing of the activities for each behavior. Once labeled, it became apparent two distinct types of behaviors were being exhibited; work and personal. Although this distinction is again qualitative in nature, based on the context of the analysis performed combined with the background information we have on the individual, this division seems sound. Behaviors having to do with personal activities are colored blue in Figure 6.30 while those having to do with work are colored green. Figure 6.31 is a re-plot of URLs visited

per day for all work related behaviors for the individual. This temporal representation of

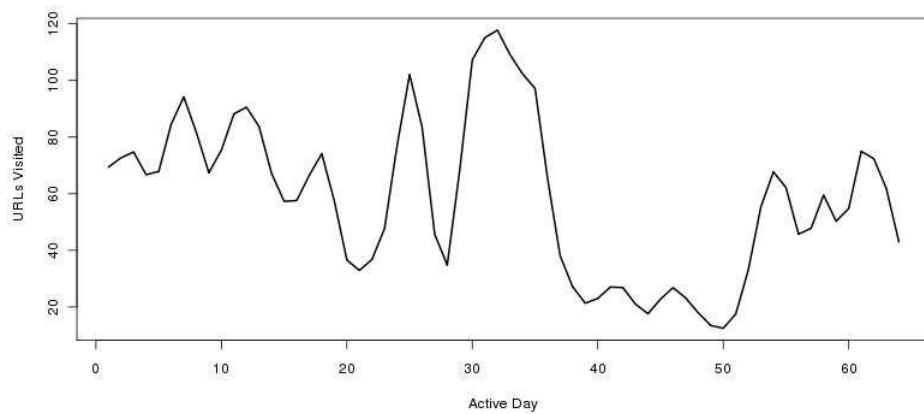


Figure 6.31: Plot of User 1's work related browsing activities.

browsing activity is significantly different than that shown in Figure 6.29, with significant increased activity occurring between days twenty five to thirty five and considerable decreases in activity from approximately day forty to day fifty. Figure 6.32 is a CUSUM chart for this data and quantitatively verifies these increases and decreases. Superimposed on this figure

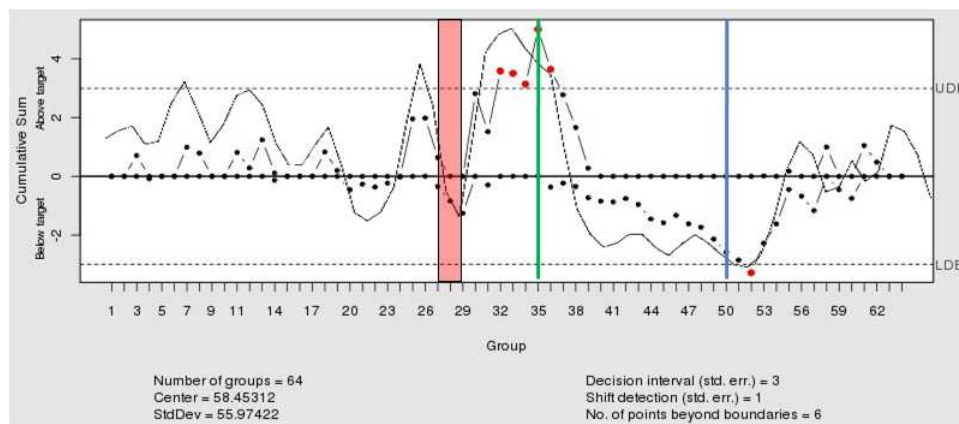


Figure 6.32: CUSUM chart of User 1's work related activities.

are known events as well as an outline of the user's work browsing pattern. From the CUSUM chart, it is obvious some sort of hyperstress related work activity is starting on or

about day thirty and continuing to day fifty four. While days forty to fifty three demonstrate below average browsing activity, as stated previously, hyperstress may show itself through significant increases or decreases in activity.

The blue line on Figure 6.32 represents the day the user defended his topic defense while the green line depicts the deadline for turning in the topic proposal to all committee members. Day twenty one is approximately two weeks prior to turning in the proposal and is where we see initial increases in work related browsing activity caused by increased focus on the coming deadline. Days twenty seven, twenty eight, and twenty nine (red bar) show a sudden steep drop off in all work activity. A simple calendar lookup shows these dates to be consistent with Thanksgiving break (Thanksgiving, the day prior, and day after). The steep drop starting on day thirty seven and continuing to day fifty three is due to the user no longer working on his topic proposal. Interestingly the user's hyperstress has not been relieved, but has simply shifted from online to offline behavior. According to the user, during this period little to no online activities were being performed as the individual was spending all of their time preparing slides for his defense. Although we have no offline data, the significant decline in online activity serves as an indicator of an event taking place.

Upon defending his proposal (day fifty), we see the user's work related activities begin to return to normal. Figure 6.33 is the CUSUM chart for the user's personal activities with an overlay of his personal browsing pattern and the previously mentioned events. The CUSUM chart in this case shows very little variation from the "norm" up until day fifty. What is interesting from day fifty on is the anomalous surge in personal related activities. Interviews with the individual about this reveals this not to be hyperstress in personal behaviors, but rather a celebratory phase associated with the proposal defense being complete. This emphasizes the importance of not only being able to distinguish user behaviors, but being able to interpret them (in this case "work" versus "play") in some way. We currently perform this activity manually and see the automated labeling of behaviors as an area of

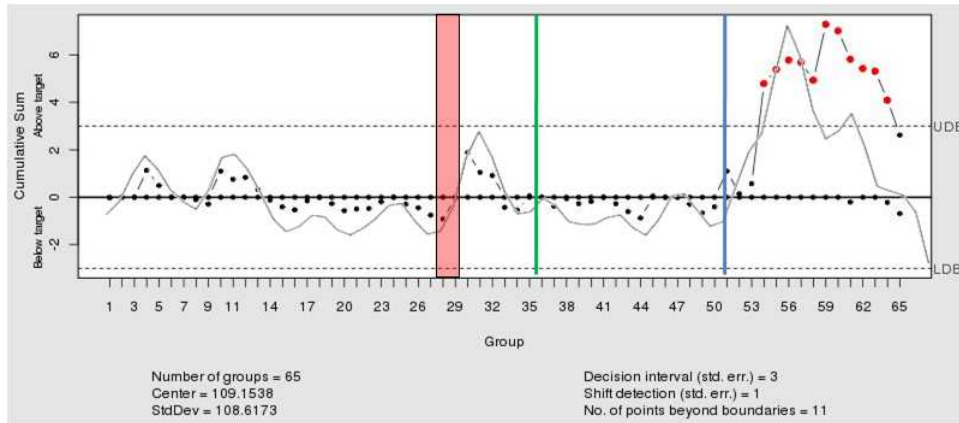


Figure 6.33: CUSUM Chart of User 1's personal activities.

future research.

Group Hyperstress

While many are not aware of the term, most individuals are familiar with the concept of group hyperstress. The Pentagon's "pizza index" or "pizza intelligence" story circulated around 1990 to 1991 and is a perfect example of how outsiders can detect extraordinary stress within a group dynamic. The story is fairly simple and is directly attributed to pizza delivery orders. Delivery people at various Domino's pizza outlets in and around Washington claimed they could anticipate breaking news at the White House or Pentagon by the upsurge in late night takeout orders. The increase was of course caused by a large number of personnel working late and needing dinner. The intensified work schedule was the stress, while the mushrooming deliveries at non-standard times was the indicator. The goal of this research is to identify similar indicators in the cyber realm, but also provide information relating to the context of the event.

As stated previously, collecting cyber-based behavioral information proved difficult due to the many privacy concerns associated with the data. Because of this, conducting a group level hyperstress experiment containing ground truth information was very difficult. Fortunately, the AOL dataset was collected covering a known high stress period for a large

number of individuals in the United States; April 15th. April 15th is the deadline for tax submission to the Internal Revenue Service in the United States. While we can only infer the individuals in question were exhibiting hyperstress, it is a fairly well known and widely accepted stressful time of year. We were able to obtain some validation of this fact using Google Trends data. Google Trends analyzes a portion of Google web searches to compute how many searches have been done for the terms entered, relative to the total number of searches done on Google over time. Using the search term “taxes” as a baseline, we can see in Figure 6.34 the annual recurring trend of this activity. Figure 6.35 is a detailed view for the

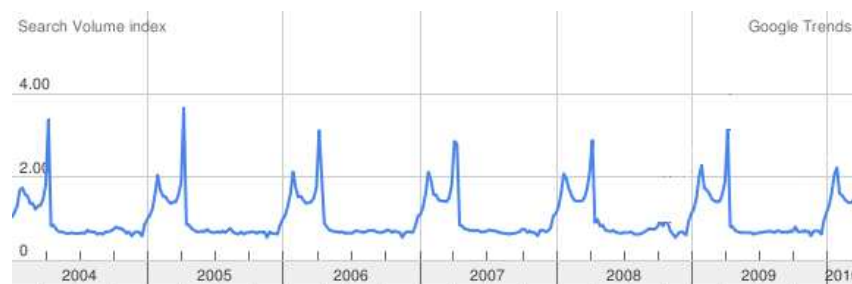


Figure 6.34: Historical plot of Google Trend data for the query term “taxes”.

year 2006 and emphasizes a definite increase in query activity for the “taxes” term around the February timeframe which then tapers off a bit before peaking just after 15 April. This is followed by a steep decline and then a relative linear trend for the remainder of the year. While the Google Trend data is informative, its usefulness is limited by knowing the term

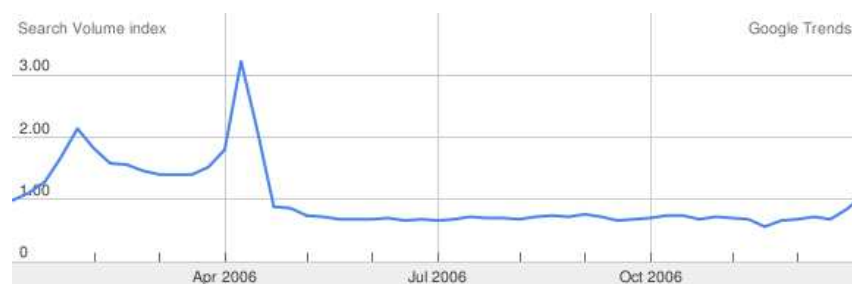


Figure 6.35: Plot of Google Trend data for the term “taxes” for the year 2006.

of interest to search for and based on that, determining if something anomalous takes place in that data. One problem with this is the large number of possible search terms associated with a given topic (in this case tax season) and all its various components. Our approach is to create a “tax profile” for behaviors of interest and then monitor this subset of data for significant change.

By querying our ontology category names and descriptions for tax related terms as well as by examining “See also” categories, we determined the most descriptive profile characteristics were categories having to do with *Internal_Revenue_Service*, *Tax*, and *Accounting*. Overall, 459 categories were related to these key terms and 1,116 users made queries in these categories during the three month period. Figure 6.36 is a graph of the queries and sessions per day for this profile. Both the Google Trend graph and our tax profile graph peak just

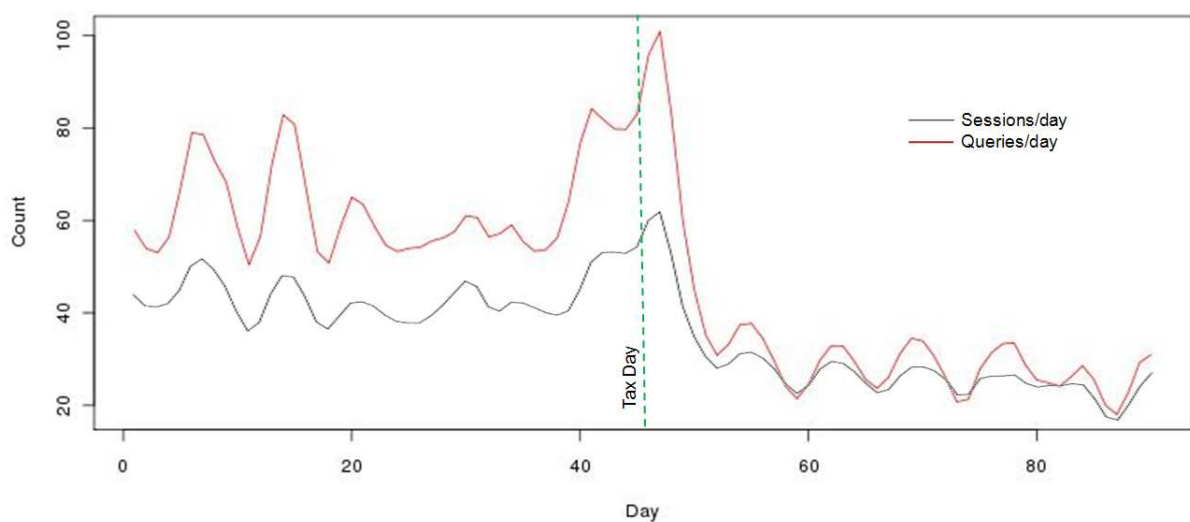


Figure 6.36: Plot of daily query and session counts for queries related to our “tax profile”.

past the 15th of April. This may be caused by late filers or those attempting to check the status of their returns. While it is easy to visualize the increases around the 7 and 15 March timeframe followed by the spike and rapid decline around the 17th of April, our goal is to detect this anomalous activity in an automated fashion. Figure 6.37 is the CUSUM for sessions

per day. This chart does an excellent job of highlighting the anomalous increases in activity

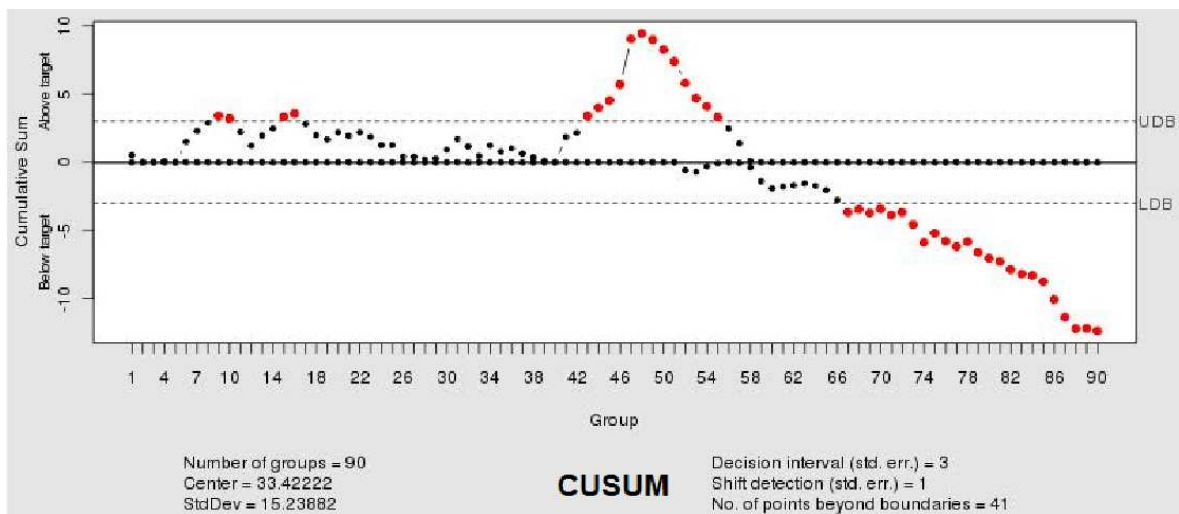


Figure 6.37: CUSUM chart of sessions/day for all queries related to our “tax profile” with significant anomalous activity detected on 15 through 17 April.

observed in early/mid March and then again mid April. Something that is a bit deceptive from using these types of charts in this situation is the apparent below average activities in this profile around the 1 May timeframe forward. Because our collection starts at the beginning of an anomalous event occurring, “normal” activity levels are initially flagged as anomalous. As more data is collected this phenomenon will automatically correct itself, but is something which must be considered when doing this type of analysis.

6.2.3 Insider Threat Detection

The last scenario we will present is in the area of insider threat detection. While there is no established formula for recognizing someone involved in espionage, certain situational factors or suitability issues have been identified which make an individual more inclined or vulnerable to exploitation by foreign intelligence officers. According to the 2002 Personnel Security Research Center (PERSEREC) study [84], “most known American spies (80%)

demonstrated one or more conditions or behaviors of security concern” before they turned to espionage. While knowing what these conditions or behaviors are is a critical first step for insider threat detection, it only addresses half the problem. The more daunting aspect involves creating a mechanism to identify, observe, and track these conditions and behaviors in a robust and reliable manner. We demonstrate how our methodology is used to mitigate this challenge. For this scenario, we will again use the AOL data set.

6.2.3.1 Behaviors of Interest

Many of the individuals who committed espionage engaged in behaviors violating the criteria for being granted an initial security clearance and for maintaining that clearance and access eligibility. These behavioral criteria are outlined in the Adjudicative Guidelines for Determining Eligibility for Access to Classified Information. Below is a subset of the behaviors identified in the guideline.

- Substance Abuse and Mental Health
 - Alcohol or Other Substance Abuse or Dependence
 - Appearing intoxicated at work
 - Irregular work schedules
 - Unexplained changes in mood
 - Decline in performance or work habits
- Inappropriate Interpersonal Or Criminal Behavior
 - Verbal or physical threats
 - Extreme or recurrent violation of rule(s) or law(s)
 - Any occasion of violence
- Finances
 - Bankruptcy
 - Reckless or compulsive spending trends, frequent gambling, or evident gambling debt

- Shortages or loss of property, sloppy handling of cash funds, disregard for financial/property administration regulations
- Unexplained or sudden large sums of cash
- Divided Loyalty or Allegiance to the U.S.
 - Strongly voiced advocacy of acts of force or violence against the U.S. Government

While not all behaviors are actionable, when occurring in combination or at severe unchecked levels, they can potentially pose a risk to the individual's well-being or to national security. Figure 6.38 is mapping of 150 insider threat cases to a number of the behaviors listed above. Again, while participation in these activities is not a definitive indicator some-

Security-Relevant Issues	Yes		No or Unknown	
	<i>n</i>	%	<i>n</i>	%
Foreign attachments	66	44	84	56
Debts that generated willingness to sell data	58	39	92	61
Illegal drug use	40	27	110	73
Immoderate alcohol use	40	27	110	73
Allegiance to a country or cause other than the U.S.	30	20	120	80
Increased spending inconsistent with known income level	27	18	123	82
Gambling	13	9	137	91
Criminal acts against property or persons, or both	11	7	139	93

Figure 6.38: Mapping of 150 insider threat cases to behaviors violating criteria outlined in the Adjudicative Guidelines for Determining Eligibility for Access to Classified Information.

one will commit espionage, Figure 6.38 empirically suggests these behaviors should not be ignored. While not demonstrated in this section, our cyber stress analysis of Section 6.2.2 would provide an alert mechanism for a number of the behaviors listed above.

For this scenario, we address the area of *frequent gambling* under the *Finances* category shown previously. Due to our focus on cyber activities, we limit our attention in this area to online gambling. Research suggests [200] electronic gambling is a more potent activity disproportionately associated with addiction when compared to traditional table games.

According to a study released in 2002, people who gamble on the Internet have more serious gambling addictions than people who wager in other ways [162].

Although our AOL data set contains queries only, we make the assumption that a user performing gambling related queries will be participating in gambling related activities.

Similar to our targeting example, the first step in the process is defining a profile to identify users exhibiting this behavior. Because we have already walked through the profile creation process in the previous two scenarios, we will simply assume a gambling profile can be created and skip to the results. The main aspects of the profile for this scenario are contextual and related to gambling activities. Two hundred and fourteen possible activities are identified meeting our criteria. Instantiating the gambling profile on the AOL data set identified 1,333 users with one or more gambling related queries. To filter our results, we added temporal constraints to our profile requiring at least one persistent profile state (states P0, P1, or P2 as defined in Section 5.5.1).

Each user is first evaluated using the contextual profile parameters and a plot of sessions versus percentage of all sessions is generated. Profile states are identified and labeled as outlined in Section 5.5.1.1. Users having at least one persistent state are added to the list of potential users. With these parameters in place, the number of users meeting the profile is narrowed to ninety eight. Due to the size of the result set, clustering is performed to separate the users into similar groups. Using our frequency based similarity measure defined in Section 5.4.3, we identify seven clusters using agglomerative hierarchical clustering. Due to the number and size of the clusters, we will only analyze one cluster in detail.

Cluster one consisted of thirty three individuals, whom we will refer to as “frequent gamblers”, and is shown in Figure 6.39. The user ids of the individuals are listed on the x axis while categories are listed on the y . The color of the squares in the figure represents the correlation between a user and an activity with red being positively correlated (darker the color, the more positive the correlation) and white being negatively correlated. To improve

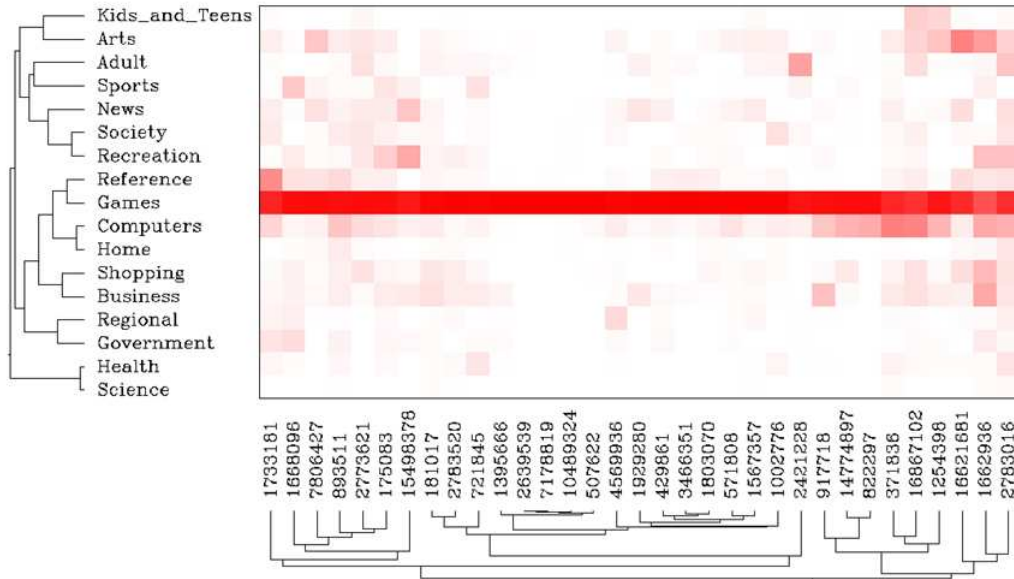


Figure 6.39: Cluster one consisting of persistent gamblers. Red squares represent a user having a positive correlation to the activity (darker the color, the more positive the correlation). The *Games* category consists almost entirely (92%) of *Games/Gambling* related queries.

readability, we only list the top level activities. The *Games* activity consists almost entirely (92%) of *Games/Gambling* related queries.

According to a 2008 national study on gambling [92], someone who gambles twice a week or more is described as a frequent gambler. Cluster one individuals far exceeded this definition. These users exhibited persistent gambling activity for not just one state, but for all sessions in which they were online. The average proportion of sessions spent on gambling related activities for these users was 89% (with four users having a 100% average over all sessions).

As our goal here is to identify users exhibiting potentially dangerous gambling behaviors, the individuals in this cluster represent appropriate candidates for further investigation. Again, although based on query activities, manual examination of the data revealed approximately 93% of the queries made by individuals in this group to be navigational in nature. The web site of interest is known to the user (i.e. Facebook, Southwest Airlines, etc.) and

the search query serves as a type of bookmark or shortcut to get there.

According to [94], approximately one to three percent of the general adult population suffers with a gambling related disorder of clinical magnitude. The thirty-three users identified in this cluster represent 0.78% of our population, and while the clinical magnitude of their problem cannot be confirmed, they represent a “group of interest”. As with our programming example, while this data does not represent ground truth, it is an indicator of the relevance of our profile.

Additional analysis on these clusters could be performed to extract pertinent behaviors, identify anomalies, or further filter based on temporal characteristics (i.e. only interested in those users gambling during work hours). As this type of analysis has already been performed in the previous two scenarios, we will not present additional results in this area.

6.3 Summary

In this chapter we demonstrated the depth and breadth of using cyber-based behavioral modeling in three disparate domains; military targeting, stress monitoring, and insider threat detection. While few verifiable data sources exist for conducting this type of experimental evaluation, the results presented are based on quantifiable analysis and are empirically justified. In the remaining chapter, we address areas of future research needed to further this domain.

Chapter 7

Future work and Conclusions

7.1 Future Work

While this dissertation provides the foundation of a cyber-based behavioral modeling methodology, we see future research in a number of areas needed to further this work to a practical and widely accepted framework. In the following sections, we outline areas of interest for future research and development.

7.1.1 Data Sources and Scalability

Perhaps one of the most pressing areas in cyber-based behavioral modeling is the acquisition/creation of verifiable data sets. While the results presented in Chapter 6 provide examples demonstrating the effectiveness of our behavioral modeling methodology, additional data sources are needed in which behaviors and activities can be verified and validated in a scientific manner. Because our research extends far beyond the engineering domain, work with behavioral scientists and psychologists is vital to ascertain the most beneficial methods and means to capture specified behavioral characteristics. Once the desired data is collected, continued behavioral domain expertise is critical to evaluate our analysis techniques

and validate our conclusions.

In order to model individuals or groups within large populations (i.e. 10,000, 100,000, 1,000,000 users), scalability is an essential issue which must be addressed further. Provided verifiable data sources exist, further research is needed to determine how well our behavioral model and associated analysis techniques will function given significant increases in data. For example, while our fingerprinting technique of Section 5.4.2 demonstrate cyber-based fingerprints characterized by browsing activity exist, it is unclear to what extent this work can be scaled. It is well accepted that user identification becomes more challenging when done amongst an expansive sampling [17]. While we have a measure to determine the minimal sample size needed to adequately model an individual, we do not currently have a corresponding measure to approximate the maximal population size.

7.1.2 Behavioral Metrics

In addition to obtaining verifiable data sources, behavioral metrics are of critical importance in this area and will require a significant amount of further research. Enumerated throughout this dissertation, many of the metrics applied require qualitative assessment by a human to determine the most appropriate attributes or measures to be instantiated. Behavioral analysis is typically perceived as belonging to the social sciences arena and relies on qualitative measures for the interpretation of human behaviors, with very little corresponding work in how cyber observables may be similarly quantified. A considerable amount of work is needed to determine metrics associated with profile attribute selection, profile accuracy (i.e. false positives and false negatives), as well as the number of behaviors most representative of a user. While a number of methods were outlined to identify the optimal number of topics within a topic model, further research is required validating the appropriateness of this measure in the behavioral domain. We again stress coordination with experts in the

social sciences as pivotal in identifying and creating methods to evaluate these metrics.

7.1.3 Behavioral Labeling

In Chapter 6 we outlined our behavioral model and explained how cyber-behaviors are extracted from the model. While the approach is quantitatively sound, it lacks a mechanism for labeling the behaviors extracted in an automated fashion. A common, major challenge in applying a topic model based approach to any text mining problem is to automatically label topics both accurately and intuitively allowing a user to interpret the discovered topic, or in our case, behavior. We see recent work done by [138][215] and others as offering viable approaches to resolve this problem, but believe further research beneficial in discovering how this will relate to the behavioral domain.

7.1.4 Behavioral State

While work presented in Section 5.5.1 on behavioral state provides a novel approach to behavioral analysis and prediction, it barely scratched the surface of the potential benefits offered by these techniques. Additional research integrating Markov processes [106] and embedded Markov chains should greatly enhance both intra and inter-state accuracies by incorporating data on the amount of time (sessions) spent in a given state before a transition occurs. Advances in sequence learning for anomaly detection [112] and sequence databases search techniques [160][3] also warrant additional examination and research. The ability to detect anomalous activities or to search for users exhibiting certain behavioral state-based characteristics would significantly enhance our work in this area.

7.2 Conclusions

In summary, this dissertation presented a novel approach to identify, extract, and analyze cyber behaviors providing the foundation for cyber-based behavioral modeling. We have defined the underpinnings necessary to support this approach through our behavioral extraction, sample size estimation, and behavioral state techniques, then empirically evaluated their use. In addition, we implemented methods to characterize, predict, and detect change in individual and group behaviors and demonstrated their effectiveness using real world data.

This work offers valuable contributions to a number of areas. Illustrated in Chapter 6 by targeting, stress monitoring, and insider threat scenarios, e-commerce, business process modeling, and parental control like applications would all benefit from an improved ability to characterize users' activities and behaviors over time, then detect changes in these behaviors based on known or unknown stimulations to the environment.

While there is no “cookie cutter” approach to any research involving behavioral modeling, this dissertation provides the foundation for analysis and interpretation of cyber-based behaviors in a quantitatively sound and repeatable manner.

Appendix A

Tools and Code Resources

A.1 Apache Commons Math

Commons Math is a library of lightweight, self-contained mathematics and statistics components addressing the most common problems not available in the Java programming language or Commons Lang. Various algorithms and data types from the Apache Commons Math library were used in accomplishing this work.

A.2 CLUTO

CLUTO is a family of computationally efficient and high-quality data clustering and cluster analysis programs and libraries, that are well suited for low- and high-dimensional data sets. All clustering performed in this dissertation was accomplished using CLUTO.

A.3 Java

Java is a programming language deriving much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is the primary language used for all behavioral modeling tasks. The results presented in Chapter 6 are all generated using Java and integrated R (see Sections A.9 and A.10).

A.4 Java Universal Network Graph (JUNG)

JUNG is an open source graph modeling and visualization framework written in Java. The framework comes with a number of layout algorithms built in, as well as analysis algorithms such as graph clustering and metrics for node centrality. JUNG is used throughout this work for both visualization and graph analysis.

A.5 JFreeChart

JFreeChart is an open-source framework for the Java programming language, which allows the creation of complex charts. JFreeChart is used to make the majority of the time series analysis graphs in this dissertation.

A.6 Lucene

Apache Lucene is a free/open source information retrieval software library, originally created in Java by Doug Cutting. It is supported by the Apache Software Foundation and is released under the Apache Software License. Lucene is used to store and access our activity ontology

as defined in Section 4.2.2.

A.7 Mallet

MALLET is an integrated collection of Java code useful for statistical natural language processing, document classification, cluster analysis, information extraction, and other machine learning applications to text. MALLET is used for behavioral extraction (Section 5.1.1) by using the LDA algorithm and calculation of the optimal number of behaviors using the HLDA algorithm.

A.8 MySQL

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. The behavioral activity database (BAD) and profile database from our behavioral modeling methodology are both implemented using MySQL.

A.9 R

R is a programming language and software environment for statistical computing and graphics. It is an implementation of the S programming language with lexical scoping semantics inspired by Scheme. The R language is widely used for statistical software development and data analysis. A number of R packages are used extensively within this work.

A.9.1 LearnBayes

LearnBayes contains a collection of functions helpful in learning the basic tenets of Bayesian statistical inference. It contains functions for summarizing basic one and two parameter posterior distributions and predictive distributions and also contains MCMC algorithms for summarizing posterior distributions defined by the user. We use the *rdirichlet* method to simulate samples from a Dirichlet distribution in our sample size estimation techniques of Section 5.3.

A.9.2 Miscellaneous Time Series Filters (mFilter)

The package implements several time series filters useful for smoothing and extracting trend and cyclical components of a time series. The Butterworth square-wave highpass filter *bwfilter* is used to smooth data prior to piecewise linear segmentation (Section 5.5.1.1).

A.9.3 Quality Control Charts (qcc)

The qcc package contains methods to create Shewhart quality control charts for continuous, attribute and count data, CUSUM and EWMA charts, operating characteristic curves, and process capability analysis. The *cusum* method is used for the creation and analysis of all CUSUM charts within this dissertation.

A.10 Rserve

Rserve is a TCP/IP server which allows other programs to use facilities of R from various languages without the need to initialize R or link against R library. The Rserve library is used to interface R from all Java code.

A.11 Weka

Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato. The Weka workbench contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality. Various algorithms within Weka were used to perform initial analysis of our data.

Appendix B

DMOZ Top Level Category Descriptions

The following top level category descriptions are extracted directly from [57].

B.1 Adult

The Adult category lists adult-oriented web sites. Included here are primarily sites with explicit sexual content.

B.2 Arts

The ODP Arts category contains English language sites about art, or “the use of skill and imagination in the creation of aesthetic objects, environments, or experiences that can be shared with others.” This includes the “liberal arts,” concerned with skill of expression in language, speech, and reasoning, and the “fine arts,” concerned with affecting aesthetics directly, and especially affecting the sense of beauty. (Quotes and paraphrases from [38]).

- Sites primarily about arts or artists (actors, painters, musicians, etc.) should be listed in the appropriate Arts category.
- Sites primarily devoted to selling products, even artwork or artists' products, should be listed in Shopping.
- Businesses that serve the arts communities without directly creating art themselves - agents, casting companies, financiers - are listed under the appropriate category of Business.
- Sites about art that is produced or viewed through computers are listed under Arts while sites about computer tools or data formats that can be used either for art or non-artistic work should be listed under Computers.
- Activities that entertain their participants more than the audience belong in Recreation.
- The study of non-artistic communication is listed under Science: Social Sciences.
- Non English language sites belong in World.

B.3 Business

The Business component of the Open Directory generally lists and categorizes English-language sites that cover business as an activity and business as an entity. Sites generally not listed in the Business category:

- Online shopping: Sites that exist primarily to sell goods and services directly to the consumer should be listed in the Shopping category. See the Shopping category description for more detail.

- Commercial sites (particularly ".coms") whose main purpose is to provide information and resources to the consumer, as an end-user of the product or services (e.g. Google, ConsumerSearch.com, XML.com, MP3.com, etc.)
- Sites that focus on providing information to consumers are listed under Home: Consumer Information. See the Consumer Information FAQ for more information.
- Local businesses: Sites for inherently local businesses – such as types of business that are commonly found in most localities and serve primarily the people and businesses in that area – are in most cases listed exclusively in Regional categories.

B.3.1 Business as an Activity

- Sites covering industrial and commercial activities involving the exchange of commodities, services, or financial resources.
- Sites offering supporting services, information and resources to business and business people, such as trade associations, educational institutions and training programs, business and economic news, events, etc.

B.3.2 Business as an Entity

- Official web sites for and about corporations and commercial enterprises (including subsidiaries) that manufacture, distribute, market and sell goods and services to other businesses (B2B) and/or consumers (B2C). Note: If the purpose of the site is to serve as a Shopping destination for consumers (or consumers and businesses), the site should be listed in the Shopping branch.
- Sites that exist solely to provide information about a company and its goods and services. Note: a company's brand name sites that focus on providing information to

consumers are listed under Home: Consumer Information. See the Consumer Information FAQ for more information.

- Sites that focus on the theoretical, practical or operational aspects of a business enterprise: accounting, finance, human resources, management, marketing, etc.

B.4 Computers

The ODP Computers category contains sites that are about:

- computers in themselves in general, such as Computer History or Ethics or Computer Science/;
- specific individual parts and areas of all or most computers, such as Hardware or Software, not primarily used for or by a single ODP top level category field;
- the use of computers for purposes spanning multiple ODP top level category fields, such as Graphics, when the graphics could be scientific or artistic or business oriented, or for databases or the Internet in general, which store or carry all sorts of information, for whatever purpose;
- fields that are only conceptually possible through the use of computer concepts, such as Artificial Intelligence or Artificial Life.

It is important not to confuse the topic with the tool or medium. Sites devoted to a subject that happens to use computers as tools or a medium should still be listed under the subject. For example, sites devoted to music, even in electronic formats, have a home in Arts/Music/. However, the tools and formats for audio reproduction may also be used for speeches or lectures that are not Arts related, so sites devoted to these generic computer audio tools,

data formats, theories, and algorithms that may be used for several purposes should be listed in the appropriate Computers subcategory.

B.5 Games

Games encompasses all English language web sites about games. The Open Directory Project defines a game as any activity that is meant to entertain the participants and that is governed by a specific set of rules. Games are also usually competitive, typically involving players competing either against each other, against one or more simulated players, or against tasks set by the rules of the game. There are usually ways to win and lose a game spelled out in the rules.

Games covers sites about nearly any sort of game, including Computer and Video Games, Board Games, Roleplaying Games, and Gambling Games, and most other activities that are commonly referred to as games. Sites about sports, recreational activities, or sites whose primary purpose is to sell items have categories specifically for their type of content.

Sports are usually more physical activities, and generally involve some form of physical exertion. Included there are sports that may be recreational rather than competitive activities for most people, such as bicycling.

Recreation is for hobbies, activities, and pastimes that are meant to give enjoyment during a participant's leisure time. These activities are generally not competitive and usually don't have a way to absolutely win or lose. They are not typically governed by a specific set of rules.

Shopping contains sites of which the primary focus is to allow the consumer to select and obtain goods and services over the web. Shopping: Toys and Games: Games is specifically for sites selling games.

B.6 Health

This section focuses on topics related to human or animal health, and medicine. Environmental health topics may be found at [Science/Environment/Environmental Health](#) . Other related categories include: [Business/Healthcare](#), [Shopping/Health](#), and [Society/Issues/Health](#).

B.7 Home

The Home category is focused on home improvement, redecorating, remodelling or do it yourself home repair, tax preparation, consumer information, gardening, apartment living, family and parenting sites, recipes or fun places for kids.

B.8 Kids and Teens

Kids and Teens is an Internet directory created especially for children and teenagers. It includes both sites designed specifically for children and/or teens as well as sites designed for general audiences. It does not include sites that are designed primarily to sell merchandise, sites that use profanity or obscenity, or sites that contain sexually explicit content.

B.9 News

News sites provide material informing, explaining, or commenting on the events and issues of the day. In addition, web sites with information 'about' news and about news reporting are within the scope of the category and its subcategories. This might include essays on current events or sites describing news sources in detail.

The Open Directory Project's news section includes a wide variety of news links. The Top: News section is a cross section of some of those links.

B.10 Recreation

The Recreation category covers English language sites about hobbies, activities, and pastimes that are meant to give enjoyment during a participant's leisure time. The purpose of the category is to provide resources and information on a multitude of recreational activities. These resources should be of far-reaching or worldwide interest and not mainly of interest to a specific regional area.

B.11 Reference

Reference is a category for sites devoted to the study of and access to information itself - teaching it (Education), using it (Knowledge Management), storing it (Libraries, Museums) - or sites containing information on topics varied and comprehensive enough that they could not be classified under the other top level Open Directory categories (Dictionaries, Directories, Encyclopedias).

B.12 Regional

The Regional category contains English language sites about geographical regions of the world.

B.13 Science

Broadly defined to include physical sciences, life sciences, social sciences, earth sciences, mathematics, engineering, and technology. The alphabetical index helps to find a particular Science topic, especially if common name differs from one used by scientists. Sites about alternative / non-mainstream Science concepts are listed in [Science/Anomalies_and_Alternative_Science](#).

B.14 Shopping

Shopping/ contains sites of which the primary focus is to allow the consumer to select and obtain goods and services over the Web. Common examples include:

- Integrated online shopping-cart systems that allow the user to order directly over the Web.
- Online shopping-cart systems that allow the user to generate an order form to be sent to the merchant via fax or mail.
- Simple directories of products and prices that the user can order via mail or phone.

B.15 Society

Open Directory's Society section covers the areas of human interaction and people's thoughts, speculations, and aspirations about the world they live in. This category covers topics directly about people, such as People, Ethnicity, Genealogy, and Disabled. And it covers experiences inherent to being human, in Sexuality, Death, Transgendered, Subcultures, Lifestyle Choices, and Gay, Lesbian, and Bisexual. Society covers people's relationships with each other in topics such as Relationships, Activism, Advice, Crime, Support Groups, Work, Military, Politics, Issues, Law, Government, and Organizations. It covers people's perceptions and understandings of the universe they live in, in Philosophy, Paranormal, Urban Legends, and Religion and Spirituality. And it covers people's relationship to time, in topics such as History, Future, and Holidays.

Several other major Open Directory sections have categories related to Society. For example, categories about people who are important to society principally because of their contributions to science, the arts, sports, or local areas are located in Science, Arts, Sports,

and Regional, respectively. Subjects related to the family are in Home/Family. The relationship between science and society is covered in Science/Science.in_Society. The Arts and Humanities, which are an important expression of society, have their own categories. News about society is in News. Many socially relevant reference materials about society can be found in Reference, such as Reference/Museums/Cultural, Reference/Archives, and Reference/Education.

B.16 Sports

Sports can be generally defined as competitive events involving physical exertion. Included here are sports that may be recreational rather than competitive activities for most people, such as bicycling.

If you don't find what you're looking for here, check the related categories Games (for competitive but non-physical activities, like chess) or Recreation: Outdoors (for outdoor physical activities that aren't normally competitive, such as hiking).

B.17 World

This category contains the non-English language versions of the Open Directory Project. Both the category headers and site descriptions should be written in the language of the sites they link to. There are over 70 languages already represented in ODP World.

Bibliography

- [1] AboutUs. Aboutus wiki page. <http://www.aboutus.org>, 2009.
- [2] A. Acquisti and R. Gross. *Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook*. 2006.
- [3] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. pages 69–84. Springer Verlag, 1993.
- [4] R. Agrawal and T. Imielinski. Mining association rules between sets of items in large databases. pages 207–216, 1993.
- [5] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
- [6] H. S. Al-Khalifa and H. C. Davis. Folksannotation: A semantic metadata tool for annotating learning resources using folksonomies and domain ontologies. November 2006.
- [7] H. S. Al-Khalifa and H. C. Davis. Folksonomies versus automatic keyword extraction: An empirical study. In *IADIS Web Applications and Research 2006*, May 2006.
- [8] S. Alag. *Collective Intelligence in Action*. Manning Publications Co., Greenwich, CT, USA, 2008.
- [9] Alexa. Alexa top 500 sites worldwide. http://www.alexa.com/site/ds/top_sites?ts_mode=global&lang=none, January 2009.
- [10] S. O. Amin, M. S. Siddiqui, C. S. Hong, and S. Lee. Rides: Robust intrusion detection system for ip-based ubiquitous sensor networks. *Sensors*, 9(5):3447–3468, 2009.
- [11] C. R. Anderson, P. Domingos, and D. S. Weld. Adaptive web navigation for wireless devices. In *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 879–884. Morgan Kaufmann, 2001.

- [12] N. O. Andrews and E. A. Fox. Recent developments in document clustering. Technical report, Computer Science, Virginia Tech.
- [13] P. G. Anick. Integrating natural language processing and information retrieval in a troubleshooting help desk. *IEEE Expert: Intelligent Systems and Their Applications*, 8(6):9–17, 1993.
- [14] Apache. Tika - content analysis toolkit. <http://lucene.apache.org/tika/>, January 2009.
- [15] B. Apolloni, A. Ghosh, F. N. Alpaslan, L. C. Jain, and S. Patnaik, editors. *Machine Learning and Robot Perception*, volume 7 of *Studies in Computational Intelligence*. Springer, 2005.
- [16] Army. Army human terrain system. <http://humanterrainsystem.army.mil/>, December 2008.
- [17] J. Ashbourn. The distinction between authentication and identification. <http://homepage.ntlworld.com/avanti>, 2000.
- [18] K. Atkinson. Gnu aspell. <http://aspell.net/>, 2008.
- [19] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, New York, NY, USA, 2006. ACM.
- [20] P. Baldi, P. Fransconi, and P. Smyth. *Modeling the Internet and the Web*. Wiley, first edition, 2003.
- [21] A. Banerjee and J. Ghosh. Clickstream clustering using weighted longest common subsequences. In *in: Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining*, pages 33–40, 2001.
- [22] P. Barnard. Infonetics report predicts continued growth for network security gear through 2007. <http://www.tmcnet.com/voip/ip-communications/articles/5360>, 2007.
- [23] S. Bausch. Two-thirds of active u.s. web population using broadband, up 28 percent year-over-year to an all-time high, according to nielsen//netratings. http://www.nielsen-online.com/pr/pr_060314.pdf, 2006.
- [24] B. Berkowitz. Learning to break the rules. Commentary, NY Times, December 2003.
- [25] A. Bestavros. Using speculation to reduce server load and service time on the www. In *CIKM '95: Proceedings of the fourth international conference on Information and knowledge management*, pages 403–410, New York, NY, USA, 1995. ACM.

- [26] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is “nearest neighbor” meaningful? *Lecture Notes in Computer Science*, 1540:217–235, 1999.
- [27] D. Blei and J. Lafferty. Correlated Topic Models. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, 18:147, 2006.
- [28] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*, page 2003. MIT Press, 2004.
- [29] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML ’06: Proceedings of the 23rd international conference on Machine learning*, pages 113–120, New York, NY, USA, 2006. ACM.
- [30] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
- [31] W. M. Bolstad. *Introduction to Bayesian Statistics*. Wiley-Interscience, 1 edition, April 2004.
- [32] C. Borgelt. Christian borgelt’s webpages. <http://www.borgelt.net/apriori.html>, January 2009.
- [33] BOTW. Best of the web directory. <http://botw.org>, 2009.
- [34] Bowling, Michael, Furnkranz, Johannes, Graepel, Thore, Musick, and Ron. Machine learning and games. *Machine Learning*, 63(3):211–215, June 2006.
- [35] T. Bradberry. *The personality code*. Viking, Camberwell, Vic. :, 2007.
- [36] M. E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, 1987.
- [37] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.
- [38] Britannica. Encyclopedia britannica, July 2009. <http://www.britannica.com/>.
- [39] BUBL. Bubl home page. <http://bubl.ac.uk>, 2009.
- [40] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Model-based clustering and visualization of navigation patterns on a web site. *Data Min. Knowl. Discov.*, 7(4):399–424, 2003.
- [41] R. Campagna. How new access control technologies can address insider threats. <http://www.scmagazineus.com/How-new-access-control-technologies-can-address-insider-threats/article/123587/>, 2009.

- [42] D. Cappelli, A. Moore, R. Trzeciak, and T. Shimeall. 2006 common sense guide to prevention and detection of insider threats. <http://www.cert.org/archive/pdf/CommonSenseInsiderThreatsV2.1-1-070118.pdf>, 2006.
- [43] L. Catledge and J. Pitkow. Characterizing browsing behaviors on the world wide web. In *Computer Networks and ISDN Systems*, volume 27, 1995.
- [44] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, volume 27, pages 307–318, New York, NY, USA, June 1998. ACM Press.
- [45] H. Chen. Bringing order to the web: automatically categorizing search results. In *Szwillus (eds), Proceedings, Conference on Human Factors and Computing Systems, The Hague*. ACM Press, 2000.
- [46] H. Chen. Dark web terrorism research. <http://ai.arizona.edu/research/terror/index.htm>, December 2008.
- [47] E. H. Chi, J. Pitkow, J. Mackinlay, P. Pirolli, R. Gossweiler, and S. K. Card. Visualizing the evolution of web ecologies. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 400–407, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.
- [48] L. Christie. Earn well, live cheap. http://money.cnn.com/2006/05/11/real_estate/men_at_telework/index.htm, 2006.
- [49] A. Cockburn and B. Mckenzie. What do web users do? an empirical analysis of web use. *International Journal of Human-Computer Studies*, 54:903–922, 2000.
- [50] comScore. comscore, inc. <http://www.comscore.com>, 2010.
- [51] Congress. Library of congress classification outline. <http://www.loc.gov/catdir/cpsol/lcco/>, July 2009.
- [52] Core. Core security technologies. <http://www.coresecurity.com/>, January 2009.
- [53] D. Cutting. Apache lucene. <http://lucene.apache.org>, 2008.
- [54] Delicious. Delicious social bookmarking. <http://delicious.com/>, January 2009.
- [55] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM Trans. Interet Technol.*, 4(2):163–184, 2004.
- [56] I. S. Dhillon, S. Mallela, and R. Kumar. Enhanced word clustering for hierarchical text classification, 2002.

- [57] DMOZ. Open directory project. <http://www.dmoz.org/>, 2008.
- [58] DoD. Joint publication 3-13 information operations. http://www.dtic.mil/doctrine/jel/new_pubs/jp3_13.pdf, 2006.
- [59] DoD. Joint publication 3-60 joint targeting. http://www.dtic.mil/doctrine/new_pubs/jp3_60.pdf, 2007.
- [60] DoD. Joint publication 1-02 department of defense dictionary of military and associated terms. http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf, As Amended Through 4 March 2008.
- [61] DoD. Joint publication 1-02 department of defense dictionary of military and associated terms. http://www.dtic.mil/doctrine/new_pubs/jp3_0.pdf, As Amended Through 4 March 2008.
- [62] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, November 2000.
- [63] R. E. User models in dialog systems. In *Stereotypes and user modeling*, pages 35–51. Springer, 1989.
- [64] N. Eagle and A. Pentland. Eigenbehaviors: Identifying structure in routine. Technical report, IN PROC. OF UBICOMP06, 2006.
- [65] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.
- [66] N. B. Ellison, C. Steinfield, and C. Lampe. The benefits of facebook "friends:" social capital and college students use of online social network sites. *Journal of Computer-Mediated Communication*, 12(4):1143–1168, 2007.
- [67] J. Erickson. *Hacking: The Art of Exploitation*. No Starch Press, 2nd edition, 2008.
- [68] M. R. et al. Insider threat study: Illicit cyber activity in the banking and finance sector, tech. report no. cme/sei-2004-tr-021. Carnegie Mellon Univ., Software Eng. Inst., 2004.
- [69] H. Eysenck. *The Biological Basis of Personality*. Transaction Publishers, February 2006.
- [70] Fack. Aol dataset mirror. <http://fack.org/AOL-user-ct-collection/>, 2010.
- [71] A. Finn, N. Kushmerick, and B. Smyth. Fact or fiction: Content classification for digital libraries. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

- [72] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Mach. Learn.*, 2(2):139–172, September 1987.
- [73] P. G., V. Karkaletsis, C. Papatheodorou, and C. Spyropoulos. Exploiting learning techniques for the acquisition of user stereotypes and communities. In *UM99 User Modeling: Proceedings of the Seventh International Conference*, pages 169–178, 1999.
- [74] E. Gabrilovich. Feature generation for textual information retrieval using world knowledge. *SIGIR Forum*, 41(2):123–123, 2007.
- [75] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 1301–1306, Boston, MA, 2006.
- [76] J. H. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artif. Intell.*, 40(1-3):11–61, 1989.
- [77] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. *Conference on Hypertext and Hypermedia*, 1998.
- [78] M. Graham. Aol dataset mirror. <http://mgraham.us/Data/AOL/>, 2010.
- [79] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proc Natl Acad Sci U S A*, 101 Suppl 1:5228–5235, April 2004.
- [80] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum. Integrating topics and syntax. In *In Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press, 2005.
- [81] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm. Dombased content extraction of html documents. In *WWW2003 proceedings of the 12 Web Conference, Budapest, Hungary*, 2003.
- [82] B. Hay, G. Wets, and K. Vanhoof. Clustering navigation patterns on a website using a sequence alignment method. In *In Proceedings of 17th International Joint Conference on Artificial Intelligence*, pages 1–6, 2001.
- [83] M. Hepp, K. Siorpaes, and D. Bachlechner. Harvesting wiki consensus: Using wikipedia entries as vocabulary for knowledge management. *IEEE Internet Computing*, 11(5):54–65, 2007.
- [84] K. Herbig. Espionage against the united states by american citizens 1947-2001. Technical Report 02-5, July 2002.
- [85] K. Herbig. Changes in espionage by americans: 1947-2007. Technical Report 08-05, March 2008.

- [86] E. Herder. Characterizations of user web revisit behavior. In *IN PROC. OF WORKSHOP ON ADAPTIVITY AND USER MODELING IN INTERACTIVE SYSTEMS (ABIS 2005)*, 2005.
- [87] T. H. Holmes and R. H. Rahe. The social readjustment rating scale,. *Journal of Psychosomatic Research*, 11(2):213 – 218, 1967.
- [88] S. Hood. Delicious is 5! <http://blog.delicious.com/blog/2008/11/delicious-is-5.html>, November 2008.
- [89] Z. Huiliang and H. S. Ying. A parallel bdi agent architecture. In *IAT '05: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 157–160, Washington, DC, USA, 2005. IEEE Computer Society.
- [90] A. Hulth, J. Karlgren, A. Jonsson, H. Boström, and L. Asker. Automatic keyword extraction using domain knowledge. In *Proceedings of the Second International Conference on Computational Linguistics and Intelligent Text Processing*, pages 472–482. 2001.
- [91] ISS. Internet security systems - leading intrusion prevention ips solutions. <http://www.iss.net/>, January 2009.
- [92] W. J. The prevalence of problem gambling among u.s. adolescents and young adults: Results from a national survey. *Journal of Gambling Studies*, 24:119–133, 2008.
- [93] A. V. Jaroslav. An overview of web data clustering practices.
- [94] D. J.L. Prevalence rates of youth gambling problems: Are the current rates inflated? *Journal of Gambling Studies*, 19:405–425(21), 2003.
- [95] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142. Springer Verlag, Heidelberg, DE, 1998.
- [96] T. Joachims. Svm-light support vector machine, 2009. <http://svmlight.joachims.org/>.
- [97] K. S. Jones and E. O. Barber. What makes an automatic keyword classification effective? *Journal of the American Society for Information Science*, 22(3):166–175, 1971.
- [98] J. Kang, J.-Y. Zhang, Q. Li, and Z. Li. Detecting new p2p botnet with multi-chart cusum. *Networks Security, Wireless Communications and Trusted Computing, International Conference on*, 1:688–691, 2009.

- [99] G. Karypis. Cluto - family of data clustering software tools. <http://glaros.dtc.umn.edu/gkhome/views/cluto/>, 2008.
- [100] G. Karypis. Cluto documentation. <http://glaros.dtc.umn.edu/gkhome/fetch/sw/cluto/manual.pdf>, 2008.
- [101] M. M. Keeney and E. F. Kowalski. Insider threat study: Computer system sabotage in critical infrastructure sectors. CERT/CC, Philadelphia, PA, 2005.
- [102] M. Kellar, C. Watters, and M. Shepherd. A field study characterizing web-based information-seeking tasks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7):999–1018, 2007.
- [103] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. In *In an Edited Volume, Data mining in Time Series Databases. Published by World Scientific*, pages 1–22. Publishing Company, 1993.
- [104] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *In ICDM*, pages 289–296, 2001.
- [105] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes. Multinomial naive bayes for text categorization revisited. In *Australian Conference on Artificial Intelligence*, pages 488–499, 2004.
- [106] M. Kijima. *Markov Processes for Stochastic Modeling*. Chapman and Hall, 1 edition, January 1997.
- [107] M. Knapp and J. Woch. Towards a natural language driven automated help desk. In *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 96–105, London, UK, 2002. Springer-Verlag.
- [108] Y. Ko and J. Seo. Automatic text categorization by unsupervised learning. In *Proceedings of the 18th conference on Computational linguistics*, pages 453–459, Morristown, NJ, USA, 2000. Association for Computational Linguistics.
- [109] A. Kobsa. Generic user modeling systems. In *User Modeling and User-Adapted Interaction*, volume 11, pages 49–63. Kluwer Academic Publishers, 2001.
- [110] V. Krebs. Social network analysis of the 9-11 terrorist network. <http://www.orgnet.com/hijackers.html>, 2008.
- [111] C. Kronberg. Shalla’s blacklists. <http://www.shallalist.de/>, 2010.
- [112] T. Lane and C. E. Brodley. An empirical study of two approaches to sequence learning for anomaly detection. *Mach. Learn.*, 51(1):73–107, 2003.

- [113] S.-j. Lin and N. Belkin. Validation of a model of information seeking over multiple search sessions. *J. Am. Soc. Inf. Sci. Technol.*, 56(4):393–415, 2005.
- [114] W. Lin, S. Alvarez, and C. Ruiz. Efficient adaptive-support association rule mining for recommender systems, 2002. <http://citeseer.ist.psu.edu/483133.html>.
- [115] W. Lin, S. A. Alvarez, and C. Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6(1):83–105, January 2002.
- [116] A. Lipsman. Comscore releases november 2008 u.s. search engine rankings. <http://www.comscore.com/press/release.asp?press=2652>, December 2008.
- [117] A. Lipsman. Cyber monday. <http://www.comscore.com/press/release.asp?press=2607>, December 2008.
- [118] A. Lipsman. Global search market draws more than 100 billion searches per month. http://www.comscore.com/Press_Events/Press_Releases/2009/8/Global_Search_Market_Draws_More_than_100_Billion_Searches_per_Month, 2009.
- [119] J. C. Lisheng. Visualizing and discovering web navigational patterns.
- [120] B. Liu. *Web Data Mining Exploring Hyperlinks, Contents, and Usage Data*. Springer, first edition, 2007.
- [121] L. Lu, M. Dunham, and Y. Meng. Mining significant usage patterns from clickstream data. pages 1–17. 2006.
- [122] R. W. Lucky. *Silicon Dreams*. St. Martin’s Press, New York, 1989.
- [123] W. Lynn. Introducing u.s. cyber command. http://online.wsj.com/article/SB10001424052748704875604575280881128276448.html?mod=WSJ_Opinion_LEFTTopOpinion, 2010.
- [124] F. Maggi and S. Zanero. On the use of different statistical tests for alert correlation - short paper. In *RAID*, pages 167–177, 2007.
- [125] G. B. Magklaras and S. M. Furnell. Insider threat prediction tool: Evaluating the probability of it misuse. *Computers & Security*, 21(1):62–73, 2001.
- [126] M. A. Maloof and G. D. Stephens. elicit: A system for detecting insiders who violate need-to-know. In *RAID*, pages 146–166, 2007.
- [127] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, July 2008.

- [128] C. Mantratzis, M. Orgun, and S. Cassidy. Separating xhtml content from navigation clutter using dom-structure block analysis. In *HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, pages 145–147, New York, NY, USA, 2005. ACM.
- [129] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(1):157–169, 2004.
- [130] A. McCallum. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, January 2002.
- [131] A. Mccallum. Rainbow, 2009. <http://www.cs.cmu.edu/~mccallum/bow/rainbow/>.
- [132] A. Mccallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. pages 786–791, 2005.
- [133] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*, pages 41–48. AAAI Press, 1998.
- [134] R. R. McCrae and J. Costa, P. C. Validation of the five-factor model across instruments and observers. *Journal of Personality and Social Psychology*, 52:81–90, 1987.
- [135] R. R. McCrae and J. Costa, P. C. Four ways five factors are basic. In *Personality and Individual Differences*, pages 653–665, 1992.
- [136] M. McGiboney. Nielsen online provides topline u.s. data for march 2009. http://nielsen-online.com/pr/pr_090414.pdf, 2009.
- [137] K. R. Mckeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, and S. Teufel. Columbia multi-document summarization: Approach and evaluation. In *In Proceedings of the Document Understanding Conference (DUC01)*, 2001.
- [138] Q. Mei, X. Shen, and C. Zhai. Automatic labeling of multinomial topic models. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 490–499, New York, NY, USA, 2007. ACM.
- [139] Merriam-Webster. Merriam-webster online - collaborate, July 2009. <http://www.merriam-webster.com/dictionary/collaborate>.
- [140] Microsoft. Microsoft adcenter labs. <http://adlab.msn.com/>, January 2009.
- [141] V. O. Mittal, H. A. Yanco, J. M. Aronis, and R. C. Simpson, editors. *Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing*, volume 1458 of *Lecture Notes in Computer Science*. Springer, 1998.

- [142] D. Montgomery. *Introduction to Statistical Quality Control*. Wiley, 5 edition, 2004.
- [143] D. Montgomery. *Applied Statistics and Probability for Engineers*. Wiley, 4 edition, 2006.
- [144] Mozilla. Mozilla firefox. <http://www.mozilla.com/firefox/>, December 2008.
- [145] S. A. Nene and S. K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE TPAMI: IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 1997.
- [146] Nessus. Nessus vulnerability scanner. <http://www.nessus.org/>, January 2009.
- [147] NetMarketShare. Browser battles. <http://marketshare.hitslink.com/report.aspx?qprid=0&qptimeframe=Q&qpssp=39>, December 2008.
- [148] Nielsen. Nielsen netratings. http://http://en-us.nielsen.com/tab/product_families/nielsen_netratings, 2010.
- [149] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. In *AAAI '98/IAAI '98: Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, pages 792–799, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.
- [150] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2/3):103–134, 2000.
- [151] NLM. Medical subject headings. <http://www.nlm.nih.gov/mesh/>, July 2009.
- [152] Nmap. Nmap - free security scanner for network exploration and security audits. <http://nmap.org/>, January 2009.
- [153] ODP. Open directory editing guidelines. <http://www.dmoz.org/guidelines/>, 2008.
- [154] D. of Defense. *Defense Acquisition Guidebook Ch. 4*. Washington, DC Pentagon, 2004.
- [155] U. B. of Labor Statistics. National occupational employment and wage estimates. http://www.bls.gov/oes/2006/may/oes_nat.htm#b00-0000, 2006.
- [156] I. Ontologies, C. Van Damme, and K. Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. In *In Proceedings of the ESWC Workshop Bridging the Gap between Semantic Web and Web 2.0*.
- [157] OPA. Online publishers association website. <http://www.online-publishers.org/>, January 2009.
- [158] S. Osinski and D. Weiss. Carrot2 clustering engine. <http://www.carrot2.org>, 2008.

- [159] V. N. Padmanabhan and J. C. Mogul. Using predictive prefetching to improve world wide web latency. *SIGCOMM Comput. Commun. Rev.*, 26(3):22–36, 1996.
- [160] S. Park, S. wook Kim, and W. W. Chu. Segment-based approach for subsequence searches in sequence databases. In *In Proceedings of the Sixteenth ACM Symposium on Applied Computing*, pages 248–252, 2000.
- [161] Patent. Us patent classification. <http://www.uspto.gov/go/classification/>, July 2009.
- [162] T. Patterson. Study: Internet gambling stakes are high. <http://archives.cnn.com/2002/HEALTH/conditions/03/17/internet.gambling/index.html>, 2002.
- [163] M. Pazzani, L. Nguyen, and S. Mantik. Learning from hotlists and coldlists: Towards a www information filtering and seeking agent. *IEEE 1995 International Conference on Tools with Artificial Intelligence*, 1995.
- [164] M. Pei, K. Nakayama, T. Hara, and S. Nishio. Constructing a global ontology by concept mapping using wikipedia thesaurus. In *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 1205–1210, 2008.
- [165] H. peter Kriegel and M. Schubert. Classification of websites as sets of feature vectors. In *IASTED International Conference Databases and Applications*, pages 127–132, 2004.
- [166] C. Pettey. Gartner says worldwide security software revenue grew 18.6 per cent in 2008. <http://www.gartner.com/it/page.jsp?id=1031712>, 2009.
- [167] P. Pirolli, J. Pitkow, and R. Rao. Silk from a sow’s ear: Extracting usable structures from the web. In *Proceedings of 1996 Conference on Human Factors in Computing Systems*, 1996.
- [168] J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict world wide web surfing. In *USITS’99: Proceedings of the 2nd conference on USENIX Symposium on Internet Technologies and Systems*, pages 13–13, Berkeley, CA, USA, 1999. USENIX Association.
- [169] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [170] J. Predd, S. L. Pfleeger, J. Hunker, and C. Bulford. Insiders behaving badly. *IEEE Security and Privacy*, 6(4):66–70, 2008.
- [171] X. Qi and B. D. Davison. Knowing a web page by the company it keeps. In *In CIKM*, pages 228–237. ACM Press, 2006.

- [172] X. Qi and B. D. Davison. Knowing a web page by the company it keeps. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 228–237, New York, NY, USA, 2006. ACM.
- [173] Quantcast. Quantcast. <http://www.quantcast.com/>, January 2009.
- [174] A. Rahman, H. Alam, and R. Hartono. Content extraction from html documents.
- [175] J. F. Rayport and B. J. Jaworski. *Introduction to e-Commerce*. McGraw-Hill, Inc., New York, NY, USA, 2004.
- [176] E. Reid and H. Chen. Mapping the contemporary terrorism research domain. *International Journal of Human-Computer Studies*, 65:42–56, 2007.
- [177] E. Reid and H. Chen. Contemporary terrorism researchers patterns of collaboration and influence. *Journal of the American Society for Information Science and Technology*, page forthcoming, 2008.
- [178] E. Rich. User modeling via stereotypes. *Cognitive Science*, 3:329–354, 1979.
- [179] E. Rich. Users are individuals: Individualizing user models. *International Journal of Man-Machine Sciences*, 18:199–214, 1981.
- [180] R. Richardson. 2008 csi computer crime and security survey. http://www.gocsi.com/forms/csi_survey.jhtml, 2008.
- [181] R. Romero and A. Berger. Automatic partitioning of web pages using clustering. In *Mobile HCI, volume 3160 of Lecture Notes in Computer Science*, pages 388–393. Springer, 2004.
- [182] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494, Arlington, VA, USA, 2004. AUAI Press.
- [183] D. Rubin. The bayesian bootstrap. *The Annals of Statistics*, 9(9):130–134, 1981.
- [184] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, December 2002.
- [185] R. Ryckman. *Theories of personality*. Thomason/Wadsworth, 2004.
- [186] G. K. S. McClure, J. Scambray. *Hacking Exposed: Network Security Secrets and Solutions*. McGraw-Hill, 2003.
- [187] W. N. Sado, D. Fontaine, and P. Fontaine. A linguistic and statistical approach for extracting knowledge from documents. In *DEXA '04: Proceedings of the Database and Expert Systems Applications, 15th International Workshop*, pages 454–458, Washington, DC, USA, 2004. IEEE Computer Society.

- [188] N. Sahoo and G. Duncan. Incremental hierarchical clustering of text documents. In *in 16th CIKM*, pages 357–366, 2006.
- [189] N. Sandell, R. Savell, D. Twardowski, and G. Cybenko. Hbml: A language for quantitative behavioral modeling in the human terrain. submitted for publication, preprint available, 2008.
- [190] J. J. Sandvig, B. Mobasher, and R. Burke. Robustness of collaborative recommendation based on association rule mining. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 105–112, New York, NY, USA, 2007. ACM.
- [191] R. R. Sarukkai. Link prediction and path analysis using markov chains. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 377–386, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co.
- [192] J. Savoy. English language stop word list. <http://members.unine.ch/jacques.savoy/clef/englishST.txt>, 2008.
- [193] J. B. Schafer, J. Konstan, and J. Riedi. Recommender systems in e-commerce. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, New York, NY, USA, 1999. ACM.
- [194] S. Schechter, M. Krishnan, and M. D. Smith. Using path profiles to predict http requests. *Comput. Netw. ISDN Syst.*, 30(1-7):457–467, 1998.
- [195] K.-M. Schneider. Techniques for improving the performance of naive bayes for text classification. pages 682–693. 2005.
- [196] P. Schonhofen. Identifying document topics using the wikipedia category network. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 456–462, Washington, DC, USA, 2006. IEEE Computer Society.
- [197] J. P. Scott. *Social Network Analysis: A Handbook*. SAGE Publications, January 2000.
- [198] A. J. Sellen and R. Murphy. How knowledge workers use the web. pages 227–234. ACM Press, 2002.
- [199] H. Selye. *The Stress of Life*. McGraw-Hill, 2 edition, 1978.
- [200] H. Shaffer. Internet gambling and addiction. <http://www.divisiononaddictions.org/html/publications/shafferinternetgambling.pdf>, 2004.
- [201] E. Shaw and L. Fischer. Ten tales of betrayal: an analysis of attacks on coporate infrastructure by information technology insiders. Defense Personnel Security Research and Education Center, Monterrey, CA, 2005.

- [202] D. Shen, R. Pan, J.-T. Sun, J. J. Pan, K. Wu, J. Yin, and Q. Yang. Q2c@ust: our winning solution to query classification in kddcup 2005. *SIGKDD Explor. Newsl.*, 7(2):100–110, 2005.
- [203] D. Shen, R. Pan, J.-T. Sun, J. J. Pan, K. Wu, J. Yin, and Q. Yang. Query enrichment for web-query classification. *ACM Trans. Inf. Syst.*, 24(3):320–352, 2006.
- [204] J. Shetty. Discovering important nodes through graph entropy: The case of enron email database. In *KDD, Proceedings of the 3rd international workshop on Link discovery*, pages 74–81. Press, 2005.
- [205] Y. Shoham. Learning information retrieval agents: Experiments with automated web browsing. pages 13–18, 1995.
- [206] V. A. Siris and F. Papagalou. Application of anomaly detection algorithms for detecting syn flooding attacks. In *In Proceedings of IEEE Globecom*, pages 2050–2054. IEEE, 2004.
- [207] Sourceforge. Html content extractor. <http://senews.sourceforge.net/>, 2008.
- [208] Sourceforge. Java text categorizing library. <http://textcat.sourceforge.net/>, January 2009.
- [209] E. Spertus. Parasite: Mining structural information on the web. *Computer Networks and ISDN Systems: The International Journal of Computer and Telecommunication Networking*, 11:1205–1215, 1997.
- [210] M. Steyvers and T. Griffiths. *Probabilistic Topic Models*. Lawrence Erlbaum Associates, 2007.
- [211] E. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *In Workshop on Artificial Intelligence for Web Search (AAAI 2000*, pages 58–64. AAAI, 2000.
- [212] A. Tanenbaum. *Computer Networks*. Addison-Wesley Publishing Company, fourth edition, 1994.
- [213] J. M. Taub. Eysenck’s descriptive and biological theory of personality: A review of construct validity. *International Journal of Neuroscience*, 94(3-4):145–197, 1998.
- [214] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in Neural Information Processing Systems*, 2004.
- [215] P. Treeratpituk and J. Callan. Automatically labeling hierarchical clusters. In *dg.o ’06: Proceedings of the 2006 international conference on Digital government research*, pages 167–176, New York, NY, USA, 2006. ACM.

- [216] URLBlacklist.com. Urlblacklist.com. <http://urlblacklist.com/>, 2010.
- [217] USJFCOM. Supplement 1, commander’s handbook for an effects-based approach to joint operations. http://www.au.af.mil/au/awc/awcgate/jfcom/ebo_handbook_2006.pdf, 2006.
- [218] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.
- [219] Vivisimo. Clusty the clustering search engine. <http://clusty.com>, 2008.
- [220] A. Vladimirov and K. Gayrilenko. *Wi-Foo: The Secrets of Wireless Hacking*. Addison-Wesley Professional, first edition, 2004.
- [221] W3C. W3c definition of common log format. <http://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format>, December 2008.
- [222] W3C. W3c definition of extended log format. <http://www.w3.org/TR/WD-logfile.html>, December 2008.
- [223] W3C. Document object model. <http://www.w3.org/DOM/>, January 2009.
- [224] Waikato. Data mining with open source machine learning software. <http://www.cs.waikato.ac.nz/ml/weka/>, January 2009.
- [225] H. Wang, D. Zhang, and K. G. Shin. Change-point monitoring for detection of dos attacks. *IEEE Transactions on Dependable and Secure Computing*, 1:2004, 2004.
- [226] X. Wang, R. Bai, and J. Liao. Chinese weblog pages classification based on folksonomy and support vector machines. pages 309–321. 2007.
- [227] X. Wang, J. T. L. Wang, K. ip Lin, D. Shasha, B. A. Shapiro, and K. Zhang. An index structure for data mining and clustering. *Knowledge and Information Systems*, 2:161–184, 2000.
- [228] L. Warwich and L. Bolton. *The Everything Psychology Book*. F and W Publications, 2004.
- [229] S. Wasserman, K. Faust, and D. Iacobucci. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, November 1994.
- [230] W. Wobcke, V. Ho, A. Nguyen, and A. Krzywicki. A bdi agent architecture for dialogue modelling and coordination in a smart personal assistant. In *IAT ’05: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 323–329, Washington, DC, USA, 2005. IEEE Computer Society.
- [231] J. Wolf. U.s. says 2008 intelligence budget was \$47.5 billion. <http://www.reuters.com/article/domesticNews/idUSTRE49R8DQ20081028>, October 2008.

- [232] Yahoo. Yahoo! directory. <http://dir.yahoo.com/>, 2008.
- [233] E. Ypma and T. Heskes. Categorization of web pages and user clustering with mixtures of hidden markov models. pages 31–43, 2002.
- [234] M.-L. Zhang and Z.-H. Zhou. A k-nearest neighbor based algorithm for multi-label classification. volume 2, pages 718–721 Vol. 2. The IEEE Computational Intelligence Society, 2005.
- [235] X.-Z. Zhang. Building personalized recommendation system in e-commerce using association rule-based mining and classification. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 7, pages 4113–4118, 2007.
- [236] G. Zoroya. Army’s suicide ‘crisis’ leads to action. http://www.usatoday.com/news/military/2010-01-28-suicide_N.htm, 2010.
- [237] A. Zubiaga, A. P. Garcia-Plaza, V. Fresno, and R. Martinez. Content-based clustering for tag cloud visualization. *Social Network Analysis and Mining, International Conference on Advances in*, 0:316–319, 2009.
- [238] I. Zukerman, D. W. Albrecht, and A. E. Nicholson. Predicting users’ requests on the www. In *UM ’99: Proceedings of the seventh international conference on User modeling*, pages 275–284, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc.